Journal of Research and Practice in Information Technology

Volume 33
number 3 2001

JOURNAL OF RESEARCH AND PRACTICE IN INFORMATION TECHNOLOGY
Formerly the Australian Computer Journal

The *Journal of Research and Practice in Information Technology* is a quarterly publication emphasising activities that have successfully connected fundamental and applied research in Information Technology with practical application. The journal publishes papers relating to both emerging research and to professional practice and thus contains articles that are of interest both to practicing information technology professionals and to university and industry researchers.

Editor (and address for correspondence):
Professor John F. Roddick
Editor-in-Chief
Journal of Research and Practice in Information Technology
Flinders University of South Australia
GPO Box 2100
Adelaide 5001, South Australia
Email: jrpit@cs.flinders.edu.au

Associate Editors
Associate Professor Graham Low
Associate Editor – Information Systems
The University of New South Wales

Professor Maria Orlowska
Associate Editor – Database Systems
The University of Queensland

Professor Bernhard Thalheim
Associate Editor – Conceptual Modelling
Brandenburg Technical University at Cottbus, Germany.

Dr Hugh Williams
Associate Editor – Computer Science
RMIT University

Denis Warne
Associate Editor – Software Engineering
BAE SYSTEMS

Michael Oudshoorn
Books Editor
The University of Adelaide

Professor Ian Witten
Associate Editor – Digital Libraries
University of Waikato, New Zealand.

Tom Worthington
Director, Publications Board
Australian National University

Office Bearers
J. Ridge, President
R. Kalkman, Vice President
P. Argy, Vice President
P. Ralston, Immediate Past President
R.G. Heinrich, National Treasurer
D.W. Furini, Chief Executive
PO Box Q534, QVB Post Office, Sydney, NSW 1230, Australia
Telephone: (02) 9299 3666
Facsimile: (02) 9299 3997
email: info@acs.org.au
URL: www.acs.org.au

Copyright© 2001, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the JRPIT's copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by the Australian Computer Society Inc.

Print Post Approved PP255003/01660

Welcome ...

to the third issue of the Journal for 2001. For this issue we have selected four papers on a wide variety of topics. Fei Li and Yan Lui discuss smoothing algorithms for MPEG video, Wei Li and Minjie Zhang look at methodologies for supporting mobile agents, Shihoon Cho, Joonwon Lee and Joongsoo Ma propose a new extension to flash memory file systems and Faye Borthick, Paul Bowen, Fiona Rohde and Meng Liu report on an experiment designed to test whether database normalisation affects the way in which business users detect anomalies in data.

Short Communications

Apart from the full papers, this issue of the Journal contains the first of what I hope will be many Short Communications. These short papers will generally be comments and insights into topical issues or short or interim results that will be of wide interest. They are limited to four pages in length and in order to ensure that they are turned around quickly, are reviewed rather than fully refereed. In this issue John Lamp discusses the Code Red virus and the "epidemic" experienced in July and August this year.

New Books Editor

I am also pleased to welcome Dr Michael Oudshoorn from Adelaide University to the editorial team as our new Books Editor. Michael succeeds David Williams who, after many years as books editor, has embarked on some exciting and very different activities and therefore decided that his term as books editor had to end. We are extremely grateful to him for his work with the Journal over the years. All book reviews should be sent to Michael at michael@cs.adelaide.edu.au.

A Third of a Century of Publication

The next issue, the last issue for Volume 33, represents a milestone in the Journal's history – one third of a century of publication. This makes the Journal one of the oldest continuously published computing journals in the world. As there will not be an editorial from me in the next issue, (it is also a special issue for other reasons), I would therefore like to pay tribute to the editors, associate editors and referees over this long period. Particular thanks should be extended to the ever-flexible Keith Collins and his team at Associated Business Publications, the production managers of the Journal since its inception in 1967. Finally, thanks should go to the Australian Computer Society for its unwavering support of the Journal over the years, particularly to Tom Worthington as Publications Board Director.

The next issue will contain a printed list of all papers published over its 34 year history; this list can also be found at the Journal's website – http://www.jrpit.acs.org.au/.

John Roddick, FACS
Flinders University

On Smoothing Algorithms for Transmission of Stored MPEG Video

Fei Li and Yan Liu

Department of Computer Science
Columbia University
New York, NY, 10027 U.S.A.
Email: Fl200@cs.columbia.edu, liuyan@cs.columbia.edu.au

In this paper, we consider smoothing algorithms in delivering stored varied-bit-rate (VBR) MPEG video over high-speed network. The required network bandwidth of a video service significantly affects its cost. Video smoothing techniques promise reduced bandwidth variablity and consequently reduced bandwidth requirement. Because of varying frame size, MPEG video introduces burstiness during transmission, when left unalleviated, will affect network management and resource utilisation. Various smoothing algorithms have been proposed to tackle bursty transmissions, giving optimal transmission schedules under various constraints. The first part of the paper gives a survey of the state-of-art smoothing algorithms for stored VBR videos. In the second part of the paper our work on satisfying each video bandwidth request under limited available bandwidth is presented.

Keywords: Smoothing algorithm, statistical multiplexing, temporal burstiness, Video-on-Demand

1. INTRODUCTION

A broad range of applications, such as Video-on-Demand (VOD), digital TV/HDTV broadcasting, and multimedia image/video database service, requires the playback of stored videos over high-speed network. Usually these media objects must be displayed on the client side at a specific time, space, and rate. For coherent reception and continuous playback at the client, strict quality of service (QoS) must be provided in an end-to-end manner. Currently, most videos are encoded using the MPEG standard (Sikora, 1997). Although encoding in MPEG reduces storage space and amortised transmission bandwidth requirement, it induces *bit-rate variations* due to variable frame size, non-casual decoding sequence and inter-frame dependency. Significant *rate variability* complicates the design of efficient real-time storage, retrieval and transport mechanisms while achieving high resource utilisation. In light of the general real-time design principle "less *bursty* (i.e. *smoother*) workloads are easier to handle." (Salehi *et al*, 1996), various smoothing algorithms have been proposed, and fall in the following three categories.

1.1 Temporal Smoothing (Work-Ahead Approach):

Work-ahead algorithms try to smooth out the source bandwidth requirement through intelligent scheduling. Based on the knowledge of the frame size and the sequence, as well as buffer size on

Copyright© 2001, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the JRPIT copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Australian Computer Society Inc.

Manuscript received: June 2000, revised May 2001
Associate Editor: John Roddick

the client side, the server prefetches video frames and heuristically deliver them to the client buffer before playback to reduce potential future burstiness. These smoothing algorithms are commonly evaluated on metrics such as, the number of bandwidth increases, total number of bandwidth changes, variability of bandwidth and peak bandwidths required (Feng and Rexford, 1997), etc. For certain situations, they provide the optimal solutions.

1.2 Spatial Aggregation (Statistical Multiplexing):

Statistical multiplexing is a *spatial aggregation* mechanism in which several individual streams are asynchronously superimposed and transported over the same channel (Krunz, 1999). The resulting aggregate traffic exhibits smoother bit rate behavior (i.e., *less variability*) than individual streams. Therefore, the bandwidth is allocated to the *aggregate traffic*, which may be much less than the sum of the per-stream allocated bandwidths. There are three different approaches with statistical guarantees, they differ in the type of source model under which the statistical multiplexing gain is evaluated. Another guarantees, *determined guarantee*, can be provided based on the *traffic stamp* proposed in (Krunz and Tripathi, 1997).

1.3 Temporal Multiplexing

Temporal multiplexing is achieved by introducing one or more per-stream buffers into the switches along the end-to-end path. We can see that this scheme is paramount to increasing the buffer capacity of the client. Such a scheme can reduce the stream's peak rate, but it is at the cost of introducing more delays before playback. In temporal multiplexing the server is not involved in the scheduling, and thus is not covered in this paper.

Although smoothing algorithm may reduce bandwidth requirement significantly, there are still moments when available bandwidth is insufficient to transmit all the videos without loss of quality. Various algorithms have been proposed to determine the tradeoffs between bandwidth utilisation and quality of picture, both before and after video encoding. Those algorithms concerning video encoding before transmission will perform frame dropping, encoding frames hierarchically etc., to reduce bit rate. They introduce distortion at different levels. Algorithms concerning stages after encoding will use rate shaping, delay introduction, etc. Rate shaping is applied during transmission when conflict happens and delay is introduced either at the beginning of playback or in the conflict interval.

The rest of this paper is organised as follows. In Section 2 and Section 3 give overviews of the organisation of MPEG video streams and a stored video transmitting system respectively. Section 4 presents state-of-the-art smoothing algorithms for stored VBR video streams in, focusing on temporal and spatial smoothing. In Section 5, we discuss some work concerning smoothing under limited bandwidth in. In Section 6, we conclude with an introduction to other works on smoothing algorithm and future research directions.

2. ORGANISATION OF MPEG VIDEO STREAMS

MPEG is a standard for compressing video and audio files. It addresses the compression, decompression and synchronisation of video and audio data. In MPEG, video and audio are compressed individually, and synchronisation is achieved through time stamps. Since video sequences contain a significant amount of *statistical* and *subjective* redundancy within and between frames, MPEG employs digital video-coding techniques to reduce such redundancy (Sikora, 1997). In this paper, we deal only with video compression.

MPEG uses a hierarchy of data structures in video encoding. Each video sequence begins with a sequence head and ends with an end-of-sequence code. Each sequence is divided into several *groups of pictures* (GOPs). Each GOP is a series of one or more *pictures*. A picture is the basic encoding unit. Each picture is made of *slices*. In general, a slice has a few *macro-blocks* and each macro-block has four *blocks* (MPEG). Figure 1 illustrates MPEG data organisation.

MPEG relies on two techniques to reduce *spatial* and *temporal* redundancies - *Discrete Cosine Transformation (DCT)* and *Block-Based Motion Estimation*. DCT coding techniques are employed on image blocks of 8 x 8 pixels to efficiently explore *spatial* relations between nearby pixels within the same image.

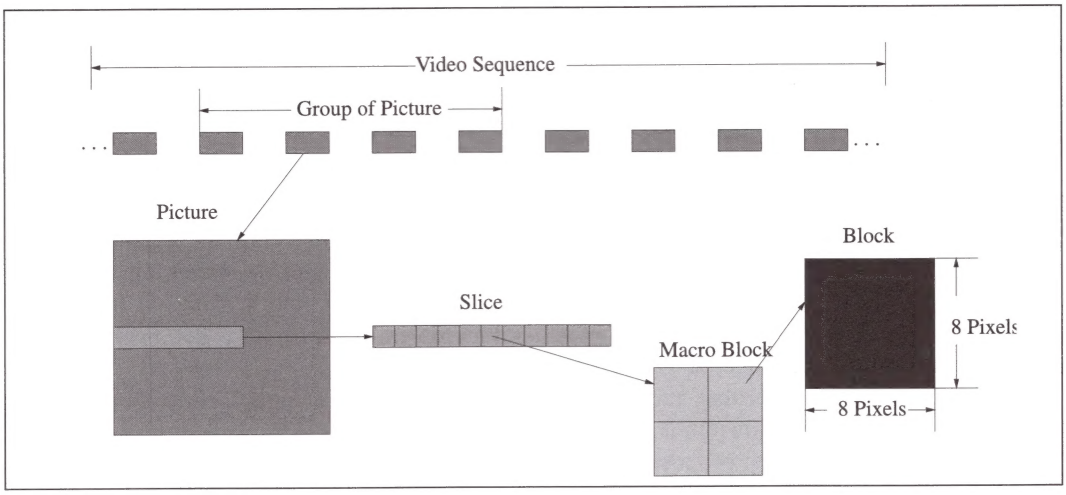


Figure 1: MPEG data hierarchy

The DCT algorithm transforms the 8 x 8 pixels from the spatial domain to the frequency domain. In the frequency domain, the 64 DCT coefficients are quantilised in a zigzag order to reduce bit rate. To achieve inter-frame redundancy, *motion estimation* technique computes *motion vectors* for every 16 x 16 macro-block. The motion vectors of a frame are predicted from its past and/or future reference frames over a search window.

In MPEG there are three types of pictures: *Intra frame* (*I*-frame), *Predicted frame* (*P*-frame) and *Bi-directional* frame (*B*-frame). *I*-frames are coded using information present only in the picture itself [MPEG]. *P*-frames are coded with respect to a past *I*-frame/*P*-frame and will be used as a reference for future predicted frames. *B*-frames are coded with reference to a past frame and a future frame and they can provide the highest degree of compression. *P*-frames and *B*-frames consist of motion vectors and/or DCT coefficients. These frames are organised into GOPs (group of picture). A GOP is led by an *I*-frame followed by *B*-frames and *P*-frames (e.g., *IBBPBBPBBPBBPBB*). Figure 2 illustrates MPEG encoding order.

There are two important facts concerning the MPEG compression standard. One is the generic structure of the MPEG standards. Users can specify the picture size and frame rate to support a wide range of application profiles. The other is that the MPEG group only standardises the decoder structure and the bit-stream formats. This allows a large degree of freedom for manufacturers to optimise coding efficiency by developing innovative encoding algorithms even after the standards

are finalised. These two properties are described by two characteristic parameters: the inter-frame to intra-frame ratio, N, and the quantiser scale parameter, q. Pancha and Zarki (1993) have shown these properties in the communication environment. They investigated ways in which a combination of varying bandwidth allocation and layered coding can be used to obtain better performance. Under limited bandwidth, a choice of these parameters is a trade-off between bandwidth utilisation and quality of picture, and results in lossy compression.

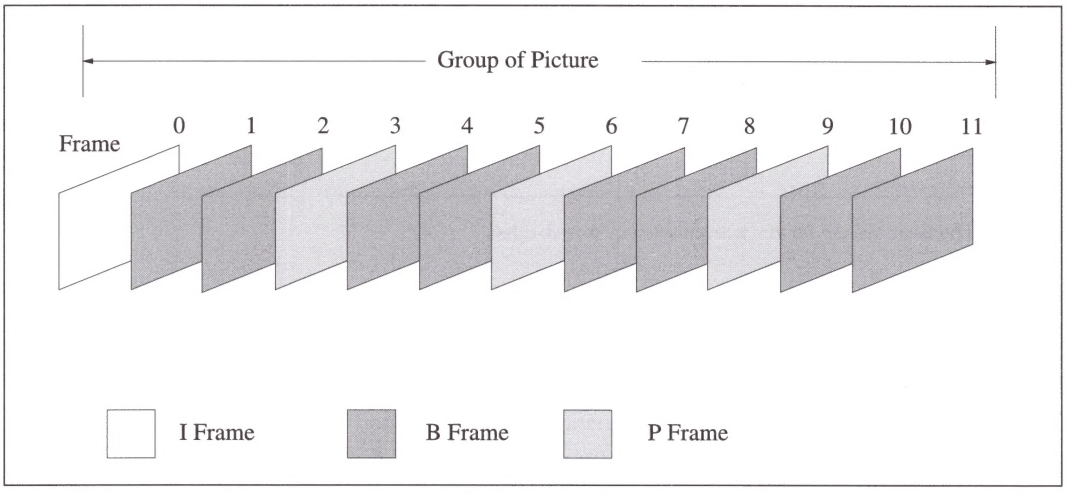


Figure 2: The arrangement of the picture coding types within the video sequence

In the subsequent sections, we use a discrete-time model $t \in \{1, 2, \ldots, n\}$ to characterise the transmission period of a MPEG video. A MPEG stream s_i is assumed to consist of n_i frames, where frame j requires f_i bytes of storage. I^i_{max}, P^i_{max}, B^i_{max} denote the maximum sizes of I, P, B frames in s_i respectively. Normally,

$$I^i_{max} > P^i_{max} > B^i_{max}$$

holds in most cases. In addition, we define L^i and Q^i which describe the GOP pattern of s_i, where, L^i is the number of frames between two consecutive I-frames in an MPEG stream; Q^i is the number of frames between an I-frame and subsequent I-frame or P-frame .

3. SYSTEM STRUCTURE FOR TRANSMITTING STORED VIDEO: AN OVERVIEW

In a generic multimedia tele-presentation system, there will be one or more media servers, a communication network and one or more clients. A set of media objects are either generated at the server from live sources, or pre-recorded in files, or a mix of both. Here, we introduce the system for stored video.

Figure 3 depicts the system structure for transmitting stored video. Video is stored at the server, and delivered to the client through network upon request. The server is responsible for the scheduling of retrieval and transmission operations. For stored video, the server has the knowledge of the frame sizes of each video as well as client buffer size.

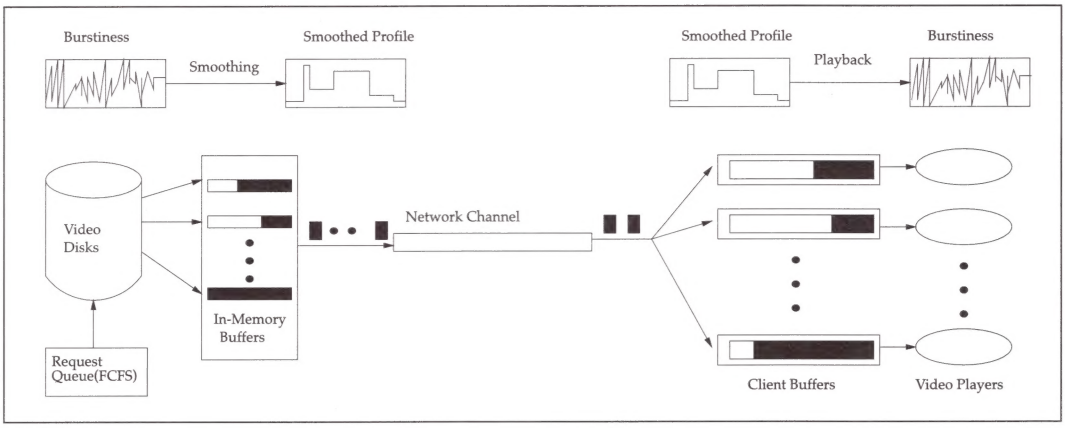


Figure 3: The basic system for the transmission of stored video.

At connection setup time, a certain amount of bandwidth is reserved. The server dynamically allocates bandwidth according to the smoothed profile of the video. Smoothing buffers are at the client sides. Frames arriving early are stored in the buffer and played back later. This duration is called *start-up delay*, which is to reduce bandwidth requirement for the initial few frames. The transmission rate through the network channel must be adjusted such that the buffer will never overflow or underflow. The system model used in this paper is a server transmits multiple, distinct VBR streams to clients across a high-speed network (Jiang and Kleinrock, 1998).

Under constant-bit-rate (CBR) network service, every video stream is allocated a fixed bandwidth, normally its peak bandwidth, for the duration of the entire transmission. This leads to a simple *static scheduling* algorithm for the video server. However, this results in the waste of bandwidth since the average playback rate is normally much lower than the peak rate used in reserving the bandwidth. Furthermore, such approach will cause client buffer overflow especially during the period of low playback rate. To avoid low network utilisation, Chen *et al* (1995) proposed a scheme that allocates a fixed bandwidth that is lower than the peak rate. As a result, during the transmission, this approach selectively discards some insignificant frames whenever the bandwidth cannot support the playback rate. Contrary to *static scheduling* approaches, *piecewise*-CBR and VBR network services offer the possibility of dynamically scheduling the transmission rate for bandwidth smoothing. CBR simplifies resource allocation and admission control procedures, while piecewise-CBR and VBR utilise resources more efficiently. To save resources and ease management, smoothing algorithms are employed in piecewise-CBR and VBR service networks. Two main smoothing approaches that do not introduce delay are work-ahead approach and statistical multiplexing.

4. SMOOTHING ALGORITHMS FOR STORED VIDEO STREAM

Smoothing algorithms can be either work-ahead approach or statistical multiplexing. The work-ahead approach is based on a stream-by-stream principle while statistical multiplexing is based on a multiple streams principle. For the work-ahead approach, smoothing algorithms have different optimisation objectives. For example, if the same bandwidth is reserved for the whole transmission, they may want to minimise the peak transmission rate or the rate variability of the smoothed video (Feng, Jahanian and Sechrest, 1997; Salchi *et al*, 1996). In a system where bandwidth can be

negotiated during the session, it is usually desirable to minimise the number of rate changes (Feng and Sechrest, 1995). For a video on demand system, it is important to support the VCR-function such as rewind, fast-forward, fast-backward etc., which can be implemented more efficiently if the requirements on the smoothing buffer and the buffer occupancy are minimised. To smooth burstiness, the server allocates a different amount of bandwidth at different time intervals for the requested video. The common feature of these algorithms is that: at each run (interval), the transmission rate can last without causing either buffer overflow or underflow. They differ in how to allocate the starting point for the next run after the rate for the previous run has been determined. Here, we give a brief review of the existing algorithms under certain constraints. These algorithms are applied to stored VBR videos.

A. Work-ahead Approach

A work-ahead approach performs temporal bandwidth smoothing on a stream-by-stream basis, assuming that the server has exact knowledge of the client buffer capabilities and the video sequence.

Consider a discrete-time model at the frame level. Suppose the video has n frames, the size of the smoothing buffer is b. To guarantee continuous playback of a MPEG stream, a server must send at least $F_{under}(t)$ cumulative bytes over [1, t] interval to avoid client buffer underflow (Feng and Rexford, 1997).

$$F_{under}(t) = \sum f_i$$

Meanwhile, the client buffer can receive at most $F_{over}(t)$ cumulative bytes over [1, t] without buffer overflow,

$$F_{over}(t) = b + \sum f_i$$

and $F_{over}(1) = b$. Consequently, a feasible transmission schedule $S(t) = \sum_{i=1}^{t} c_i$ must satisfy

$$F_{under}(t) \leq S(t) \leq F_{over}(t)$$

where c_i is the cumulative data transmitted by the server over [i - 1, i]. The area bounded by F_{over} and F_{under} is a feasible area to schedule the transmission. Any feasible schedule, as shown in Figure 4, in this area will not cause buffer underflow and overflow problems.

Obviously, there exists a set of feasible schedules, Γ, for the transmission of the MPEG streams. To video server researchers, the problem is how to find a feasible schedule, $S^* \in \Gamma$,which is as smooth as possible. The peak rate and variance of a schedule S are defined as,

$$peak(S) = \max f_i$$

$$var(S) = \frac{1}{n}\sum_{i=1}^{n}(f_i - f)^2$$

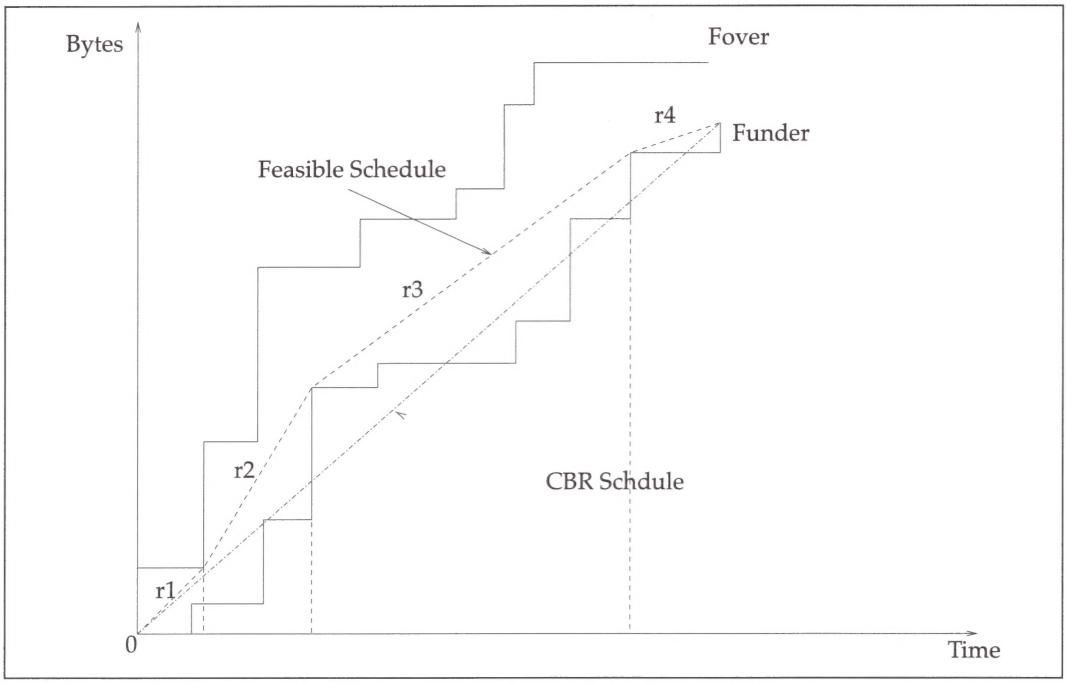


Figure 4: Transmission Schedule

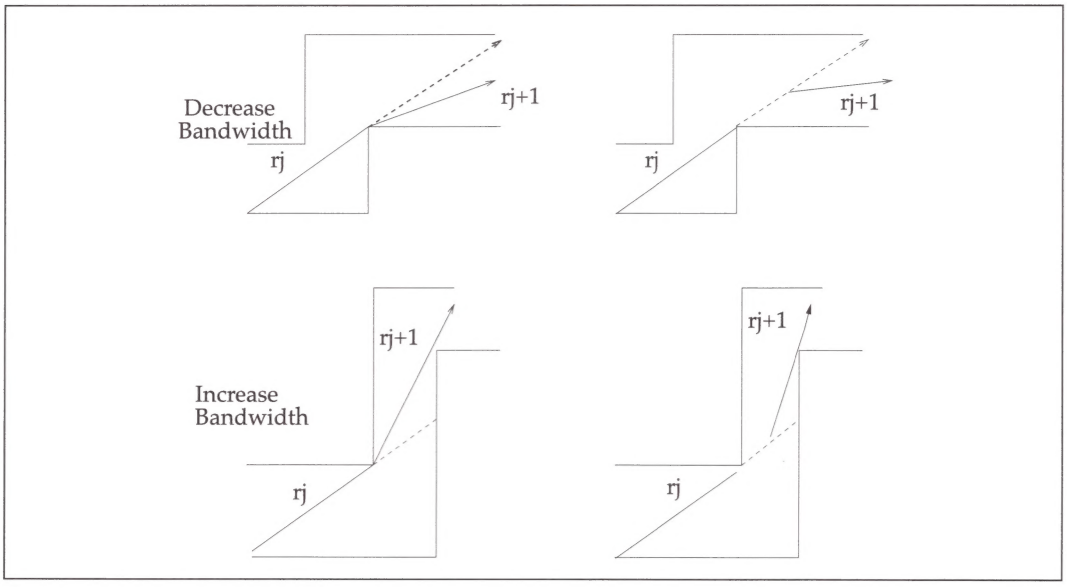


Figure 5: Allocation new bandwidth

where $f = \frac{1}{n}\sum_{i=1}^{n} f_i$. An optimal schedule S^* should have minimum peak rate and variation, i.e., $peak(S^*) \leq peak(S)$ and $var(S^*) \leq var(S)$ for any schedule S. Notice that a CBR-schedule $S = [f, \ldots, f]$ is optimally

smooth, however, the CBR schedule in Figure 4 is infeasible, and suffers from buffer underflow during certain periods.

Creating a feasible schedule, in general, involves creating m consecutive runs. In Figure 4, the feasible schedule has $m = 4$. These runs join together to form a monotonically increasing *piecewise-linear* path; each run r_j will allocate $\frac{dr_j}{dt}$ bandwidth for a duration. The smoothing algorithms select the starting point for run $r_j + 1$ based on the trajectory of run r_j. For instance, in Figure 4, if r_2 does not have its rate decreased, the schedule will cause buffer overflow. Therefore, the starting point of r_3 is selected with $\frac{dr_3}{dt} < \frac{dr_2}{dt}$.

Figure 5 illustrates the examples. Basically, there are two selection schemes,

1. Locate a starting pointing for $r_j + 1$ as early as possible before the trajectory of r_j intersects F_{over} or F_{under} (Salehi *et al*, 1996).

2. Search along the trajectory of r_j to locate the starting point that allow $r_j + 1$ to extend as long as possible (Feng, Jahanian and Sechrest, 1997)

The former approach minimises the variability of bandwidth and peak rate with $O(n^2)$ running time; while the latter approach minimises the number of bandwidth changes, m, with $O(n^2 logn)$ run time.

Feng and Rexford (1997) compared the performance of several smoothing algorithms: critical bandwidth allocation (CBA) (Feng and Sechrest, 1995), minimum change bandwidth allocation (MCBA) (Feng, Jahanian and Sechrest, 1997), and minimum variability bandwidth allocation (MVBA) (Salehi *et al*, 1996). For any rate changes, the CBA algorithm starts a rate decrease at the longest trajectory for run j+1, based on the selected starting point and initial buffer occupancy, and chooses the starting point as early as possible when it is to increase bandwidth. This results in a transmission plan that has the smallest possible peak bandwidth requirement and the minimum number of bandwidth increases. MCBA extends the CBA scheme that choose the trajectory point as far as possible regardless of increasing or decreasing bandwidth requirement. This results in a transmission plan with the smallest possible number of rate changes and minimum peak bandwidth requirement. MCBA minimises m resulting in saving of the cost of renegotiating bandwidth requirement across the network, but the saving is at the expense of increased size of each request. Salehi *et al* (1996) have proved that MVBA can schedule an optimal transmission plan, which minimises the peak rate requirement and variance of bandwidth. As a result, MVBA is able to gradually alter the bandwidth requirement. In practice, nevertheless, the *work-ahead approach* is limited by the requirement of real-time responses to the changes of a client buffer, and to the interactive VCR functions. MVBA approach minimises the variability of bandwidth and peak rate with $O(n^2)$ running time; while CBA and MCBA approaches minimise the number of bandwidth changes, i.e., minimise m, and the peak bandwidth requirement with $O(n^2 logn)$.

In contrast to CBA, MCBA and MVBA, the *piecewise constant rate transmission and transport* (PCRTT) algorithm creates bandwidth allocation plans by dividing the video stream into fixed-size intervals. This $O(n)$ algorithm generates a single run for each interval by connecting the intersection points on the F_{under} curve, as shown in Figure 6. The slopes of these lines correspond to the rates r_j in the resulting transmission plan. To avoid buffer underflow, the PCRTT scheme vertically offsets this plan until all of the runs lie above the F_{under} curve. Raising the plan is equivalent to introducing an initial playback delay at the client; the resulting transmission curve also determines the minimum acceptable buffer size to avoid overflow given the interval size, as shown in Figure 6. Because the PCRTT algorithm determines the buffer size from the bandwidth plan, it can be difficult to calculate an appropriate interval size given a fixed buffer b (McManus and Ross, 1996). An extension to the

PCRTT algorithm (PCRTT-DP) employs dynamic programming to calculate a minimum-cost transmission plan that consists of m runs. Focusing on the buffer size b as the cost metric, PCRTT-DP iteratively computes the minimum-cost schedule with k runs by adding a single rate change to the best schedule with $k - 1$ rate changes. This algorithm has a time complexity of $O(n^3)$.

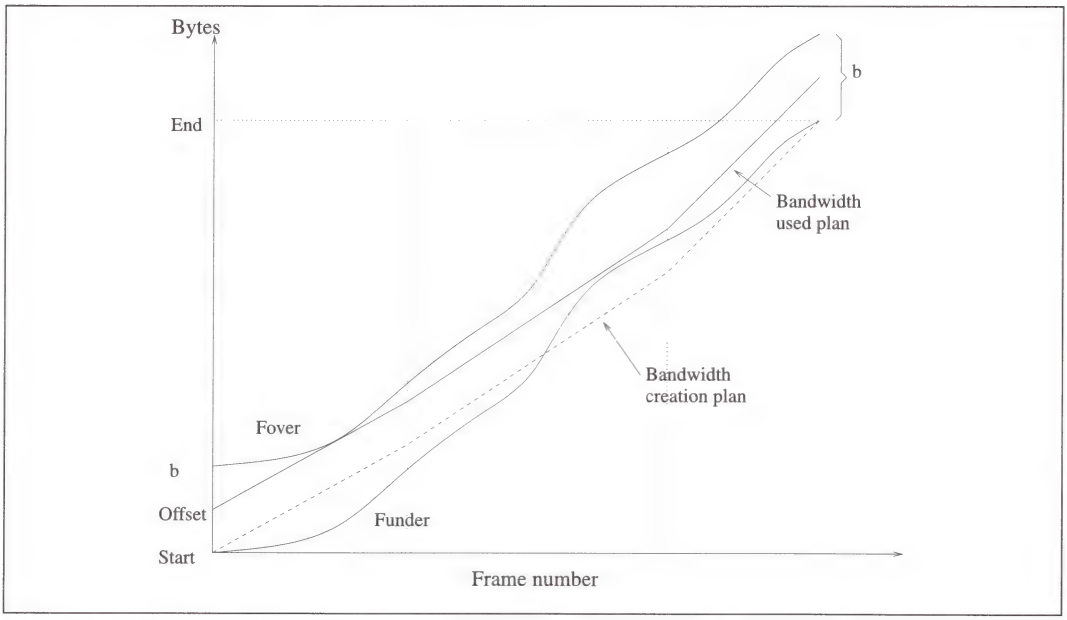


Figure 6: PCRTT plan creation

Table I summaries these five algorithms under several metrics. For more information about the metrics and their constraints on applications, please refer to Feng (1997). In this table, "Yes" means the algorithm optimises the corresponding metric, "No" means otherwise.

Algorithm	Peak Rate	Variability in Bandwidth	Number of Bandwidth Changes	Periodicity	Buffer Utilisation	Running Time (Worst Case)
MVBA	Yes	Yes	No	No	Near 50%	$O(n^2)$
MCBA	Yes	No	Yes	No	Near 50%	$O(n^2 logn)$
CBA	Yes	No	No	No	Lowest	$O(n^2 logn)$
PCRTT	No	Largest	No	Yes	Highest	$O(n)$
PCRTT-DP	Near MCBA	No	No	No	Near 50%	$O(n^3)$

Table I: Performance of five algorithms in spatial smoothing

B. Statistical Multiplexing

Statistical multiplexing (SM) is a mechanism to reduce the bandwidth requirement of bursty and VBR traffic sources (Krunz, 1999). It is being used in ATM networks in conjunction with QoS

guarantees. In essence, statistical multiplexing is a spatial aggregation mechanism in which several individual streams are asynchronously *superimposed* and transported over the same channel. The resulting *aggregate traffic* exhibits smoother bit rate behaviour, i.e., less variability, than the original streams. Bandwidth is allocated to the aggregate traffic, resulting in a reduction in the per-stream allocated bandwidth. An example of statistical multiplexing mechanism is shown in Figure 7.

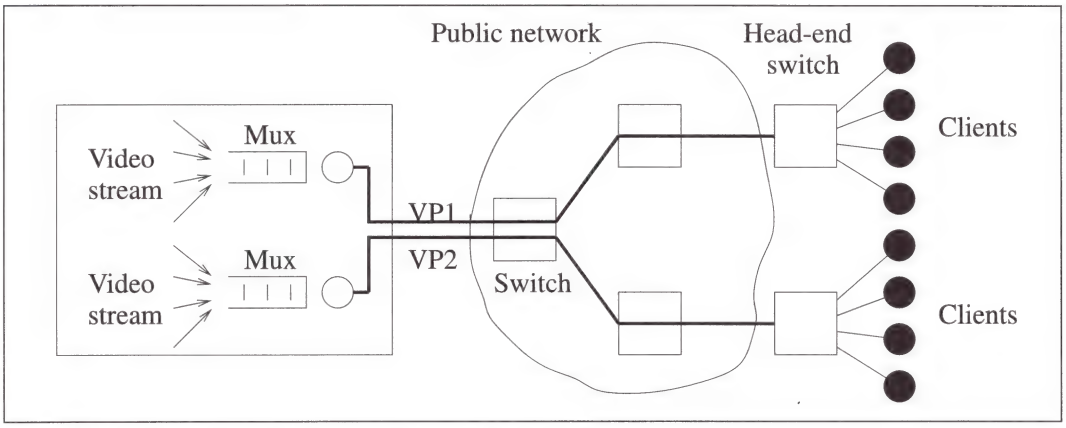


Figure 7. Example of statistical multiplexing at a video server

Traditionally, SM is used inside network switches and routers. It can also be employed at end system. For example, a remote video server may use SM to maximise the number of simultaneous video connections transmitted from the server to a head-end (HE) switch over a fixed-bandwidth pipe, see Figure 7. In this scenario, multiplexing is performed among streams that are destined to the same head-end switch, which in turn demultiplexes the traffic into several streams.

Other scenarios for SM are possible. For example, the distribution network may consist of several local servers in addition to a central remote server that maintains a repository of compressed videos. A subset of these videos is periodically multicast to the local servers. Each local server is connected to a head-end switch that serves a pool of clients. In this scenario, SM can be performed over the high-speed link connecting the remote server to each local server, and also over the link between a local server and the associated HE switch (Krunz, 1999).

Two guarantees called *statistical guarantee* and *deterministic guarantee* are provided in the statistical multiplexing mechanism.

Three different approaches can be used to provide *statistical guarantees* through statistical multiplexing. They differ in source models used in evaluating statistical multiplexing gain.

1. The first and most common model one uses is the *stochastic source models*. This model is useful in assessing the transport performance for a video whose *traffic profile* is not known at the time of admission control. Based on the queuing analysis of a video *traffic model*, the server determines the minimum amounts of bandwidth and buffer needed to guarantee a certain level of QoS. Bandwidth gain is attained by allowing the sum of the peak rates of the input streams to exceed the service rate of the multiplexer. This model can be classified according to the persistence of their auto-correlation structure into four models. They are: renewal model (no correlations), Markovian and autoregressive models (exponentially decaying autocorrelation), sub-exponential models (the decay of the auto-correlation is slower than exponential but faster than a power function) and long range

dependent (LRD) models (autocorrelation decays as a power function). Interested readers can explore more in Izauierdo and Reeves (1999).

2. Instead of detailed stochastic models, video sources can be specified by *stochastic bounds*. For example, if $X(t)$ is the number of cell arrivals in any interval t, a stochastic bound could be given by

$$Pr[X(t) > \theta_t] = \varepsilon_t$$

for several values of t, θ_t and ε_t. End-to-end guarantees are obtained by first obtaining stochastic bounds on the traffic at the edge of the network, which are then used to limit the ongoing traffic from that node. In turn, the bounded departure traffic of one node is used to bound the arrival traffic at the next node, and the procedure is repeated for all nodes along the path. A drawback of this approach is that the bounds become loose as more nodes are traversed.

3. In contrast to the above two approaches, a video source can be characterised by *deterministic time-invariant traffic envelope*. When several of these traffic envelopes are statistically multiplexed, the multiplexing gain depends on the relative phase shifts of these envelopes. Since the envelopes are time-invariant, their phase shifts can be assumed to be random, giving rise to statistical QoS guarantees.

Statistical multiplexing with *deterministic guarantees* was proposed by Krunz and Tripathi (1997). They used *traffic-envelope model* to smooth the delivery of VBR MPEG streams. The traffic-envelope of a stream s_i is a *time varying periodic* function $b_i(t)$ characterised by

$$E^i = (I^i_{max}, P^i_{max}, B^i_{max}, L^i, Q^i)$$

where the first three parameters denote the maximum size of *I*-frame, *B*-frame, *P*-frame in s_i, and the last two parameters describe the GOP pattern of s_i (Section 2). Under the above traffic envelop characterisation, MPEG sources can be statistically multiplexed at the video server and transported to a head-end switch with minimal bounded delay and no losses. The server maintains an active table of the traffic envelopes of all ongoing videos. These envelopes are updated when a new stream is added or an ongoing one is terminated. $U = [u_1, u_2, \ldots u_N]$ is described as the schedule start-up times of multiple streams at the network switch. The effective bandwidth of N streams is defined as Ng and Song (1997),

$$C = \frac{1}{N} \max\{\sum_{i=1}^{N} b_i(t - u_i)\}$$

where $t - u_i$ is an index to the traffic envelope of stream s_i and $u = [u_1, u_2, \ldots, u_N]$ describes the schedule of multiple streams at the network switch. Suppose all the streams are homogeneous ($L_i = L$ for all i) and all traffic envelopes are the same. Then

$$C = \frac{1}{N}\{m * I_{max} + n * P_{max} + k * B_{max}\}$$

where m, n, k are the number of frames belongs to *I*-frame, *P*-frame, and *B*-frame respectively, and $m + n + k = N$. If Equation (1) holds, then

$$C \leq I_{max}$$

which means that the effective bandwidth is often less than the peak rate. Krunz and Tripathi (1997)

further proved that there exists an optimal scheduling u^* which gives minimal C^* if all streams are homogeneous.

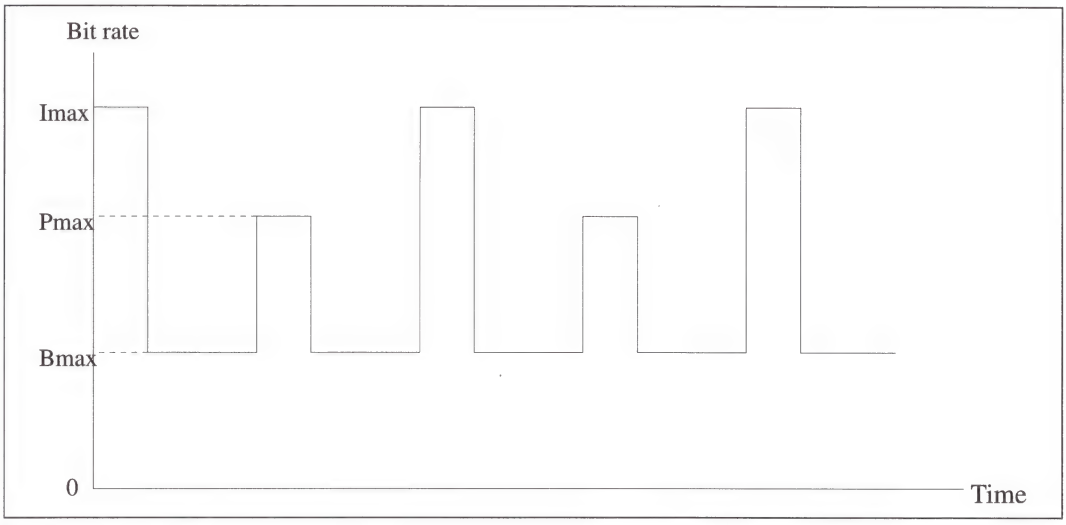


Figure 8. Traffic envelope with L=6 and Q=3

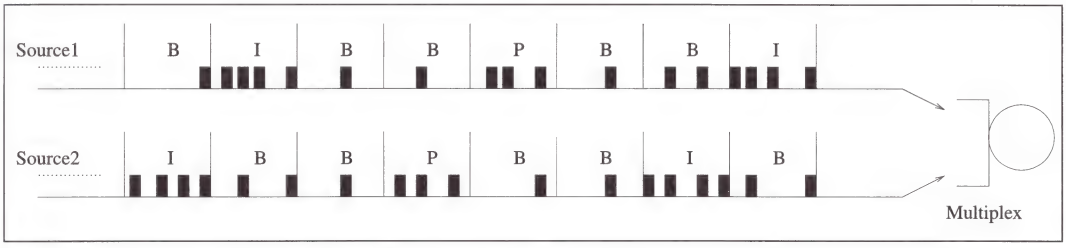


Figure 9. An example of statistical multiplexing of two MPEG sources

C. Hybrid Approach

In general, integrating work-ahead approach and statistical multiplexing can offer a certain degree of improvement in performance. In Zhang *et al* (1997), investigated the statistical multiplexing gain of a work-ahead schedule. They first conducted experiments to study the effect of client buffer size on the work-ahead schedule. As expected, larger buffer size produces smoother schedule. Then, they investigated the impact of a work-ahead schedule on *auto-co-relation* structure. A smoother work-ahead schedule has higher *auto-co-relation segments* (runs). Empirical results indicate that potential statistical multiplexing gain may diminish if the video streams are high-co-related. This implies: the hybrid approach is more suitable for clients with smaller buffer size, the statistical multiplexing gain diminishes if most of the clients display the same movies during a certain period.

5. TRANSMISSION UNDER LIMITED BANDWIDTH

In the realm of video-on-demand, video usually has to be transmitted under limited bandwidth. Methods employed can be classified using delay and distortion they introduce. In general, a compromise between quality of picture and delay is needed to satisfy the limitation of bandwidth.

A. Without Delay or Distortion

Feng (1997) proposed *rate-constrained bandwidth smoothing* (RCBS) to deal with rate constraint. It is for VCR functions for interactive video-on-demand system. Given a maximum rate constraint r, any frame that exceeds the rate constraint is modified to the maximum rate constraint and then pre-fetched. To minimise the peak rate requirement, as well as to maximise buffer utilisation, RCBS first uses $O(n)$ MVBA to determine the smallest possible peak rate for a given buffer size. Then, this rate constraint is used in computing the RCBS schedule. Given the fixed maximum bandwidth rate, this algorithm minimises the smoothing buffer residency times, the clients and servers can remain more tightly coupled making VCR functions easier to support.

In general, the server transmits requested video according to pre-computed smoothed profile. Under the circumstances that a conflict happens between the bandwidth profile and the smoothed profile, the server has to re-compute the smoothed profile, which will incur a lot of real-time computation. In Liu, Lee and Li (1999), presented a method to test conflict intervals and update the original smoothed profile. Instead of running smoothing algorithm, smoothed profile is updated using the current bandwidth profile at the conflict intervals. The limitation of this method is that it is only applied when bandwidth is sufficient for strict playback but conflict happens between smoothed profile and current bandwidth profile.

B. Without Delay and With Distortion

Transmission without delay can be achieved through introducing distortion. Three methods are used in introducing the delays. The first one is to transmit a subset of the streams if it has been scalably compressed. This is based on the characteristic of MPEG compression method. The second is to delete some less important frames, which can be achieved by subjective redundancy. Ng and Song (1997), modified the algorithms proposed in Salehi *et al* (1996) and proposed two schemes in dealing with transmission under limited bandwidth. One of the two schemes is to delete some frames. For the smoothing of MPEG-1 videos, they prefer to delete B-frames based on previous experiments that show subjective effects without loss of quality. If the bandwidth required still exceeds the bandwidth available, they will delete *PF* also. The third scheme is to insert a rate shaper between the encoder and the communication network. The rate shaper re-quantise bit rate dynamically according to the bandwidth constraints. This work is to find the optimal breakpoint during the zigzag quantisation. Methods for finding the optimal breakpoint can be found in Eleftheriadis and Anastassiou (1995).

Tse and Liew (1995) considered the degradation of image quality when the network can only support an aggregated bandwidth **A** which is less than the total requested bandwidth. Let **A** be the number of bits in a video stream s_i that must be retained in order to maintain an image distortion level D, they optimise **A** by distorting all images to the same degree. Namely,

$$\sum_{i=1}^{N} A_i(D') = \mathbf{A}$$

such that all s_i are guaranteed to have the same distortion D' as a result of insufficient bandwidth. Their proposed approach progressively discards insignificant DCT coefficients while aggregately multiplexing video streams until equality of Equation 12 is reached. In practice, it may not be possible to achieve absolute equality of distortion level because of the discrete nature of bits or groups of bits that are dropped.

C. With Delay and Without Distortion

Delay can be introduced at the beginning or at the conflict interval during transmission to preserve quality under limited bandwidth. Li and Liu (1999) proposed a fast algorithm in deciding minimal delay at the beginning of playback to satisfy limited bandwidth without loss of quality. They chose safe intervals for the runs of the smoothed profile of the requested video. The intersection of all safe intervals makes the safe interval for the transmission. The delay chosen is the smallest value in the intersection, which returns the shortest delay.

The delay can also be introduced at the conflict intervals. In Salehi *et al* (1996), only when a conflict happens, are streams paused.

Case 1: If the last segment transmits at the highest feasible constant rate, the client plays back continuously until the buffer is empty. Then client pauses for a negotiated time and starts the next segment.

Case 2: If the last segment transmit at the lowest feasible constant rate, the client pauses for a negotiated time and then starts requesting the next segment.

Jiang and Kleinrock (1998) provides a promising work. A model with parameters describing the performance of the video service is provided. At first, the performance is represented as nodes with each node denoting the performance of an interval during transmission. Then, with the constraints and changes of the environment, the server will search among the nodes for the optimal performance. That is, the server tries to minimise the cost in transition. Costs are tagged on the arcs from one node to another. Thus, finding the transition between points is converted to finding the shortest path. Jiang and Kleinrock (1998) used dynamic programming to find the best way through the points. But since the number of nodes are limited, they may not be the optimal all the time. The number of nodes needed in the describing system depends on the constraints of the system.

6. CONCLUSION

In current video applications, more smoothing needs to be done to satisfy different constraints. For Video-on-Demand systems, we have to consider operations such as forward, backward, resume and stop during strict playback. These operations require smoothing with the least buffer utilisation.

Based on practical considerations of heterogeneous and dynamically variable client buffer sizes, variable worst-case network jitter estimates, and client interactivity, Rao and Raghavan (1999) presented an algorithm for pre-computing and storing the optimal schedule for all possible client buffer sizes to satisfy an on-line computation of the schedule of transfer. Many other works are done in smoothing stored video steams, such as Koprulu, Meliksetian and Chen (1997), which uses thresholds to indicate the current occupancy of the buffer and decide whether to increase or decrease the bandwidth.

Many of the studies reviewed here have shown that the schedule must provide an intelligent algorithm to satisfy requests at some QoS level, especially adapting the changing environment constraints. Till now, most studies have been focusing on the server side. A few topics for future research are,

1. The work-ahead approach assumes that the server has the prior knowledge of the client buffer size. But it is not always true in practice since the buffer available for video applications varies. Thus, how to adjust the schedule at the client's side in order to get better performance?

2. Statistical multiplexing is used in many environments. However, under limited bandwidth, it is not preferred, as it cannot reduce bandwidth requirement. Therefore, how to adapt the residual bandwidth to save transmission time and resources?

3. All the algorithms target at finding the best tradeoffs between QoS and utilisation of resources. If hierarchical structure is used, how to make good scheduling between these QoS level?
4. Some researchers attempt to predict the bandwidth requirement to smooth video. Based on the current and past display and video size, the server predicts the upcoming traffic and adjusts the bandwidth requirement, however, this requires more statistical modelling. As a result, modifying the concept of predicting bandwidth requirement to adapt to various environments is a difficult issue.

REFERENCES

CHEN, H. J., LITTLE, T. D. C., and VENKATESH, D. (1995): *A storage and retrieval technique for scalable delivery of MPEG-encoded video*. Journal of Parallel and Distributed Computing, 30(2): 180-189.

ELEFTHERIADIS, A., and ANASTASSIOU, D. (1995): *Meeting arbitrary QoS constraints using dynamic rate shaping of coded digital video*. Proceedings of 5th International Workshop on Network and Operating System Support for Digital Audio and Video, Durham New Hampshire, April 95-116.

FENG, W. (1997): *Buffering techniques for delivery of compressed video in video-on-demand system*. Kluwar Academic Publishers.

FENG, W. (1997): *Rate-constraint bandwidth smoothing for the delivery of stored video*. SPIE Multimedia Computing and Network.

FENG, W., JAHANIAN, F., and SECHREST, S. (1997): *An optimal bandwidth allocation strategy for the delivery of compressed pre-recorded video*. Multimedia System, 5(5): 297-309.

FENG, W., and REXFORD, J. (1997): *A comparison of bandwidth smoothing techniques for the transmission of pre-recorded compressed video*. IEEE INFOCOM'97, Kobe, Japan, April, 58-66.

FENG, W., and SECHREST, S. (1995): *Critical bandwidth allocation for delivery of compressed video*. Computer Communication, October, 18: 709-717.

IZAUIERDO, M. R., and REEVES, D. S. (1999): *A survey of statistical source models for variable-bit-rate compressed video*. Multimedia Systems, July, 3: 199-213.

JIANG, Z. M., and Kleinrock, L. (1998): A general optimal video smoothing algorithm. IEEE INFOCOM'98, Proceedings, 2: 676-684.

KOPRULU, T., MELIKSETIAN, D., and CHEN, C. Y. R. (1997): *Smoothing algorithms for the delivery of compressed video*. Department of Electronic Engineering and Computer Science, Syracuse University.

KRUNZ, M. (1999): *Bandwidth allocation strategies for transporting variable bit rate video traffic*. IEEE Communication Magazine, January, 37(4): 40-46.

KRUNZ, M., and TRIPATHI, S. K. (1997): *Impact of video scheduling on bandwidth allocation for multiplexed MPEG streams*. ACM Multimedia Systems Journal, 5: 347 - 357.

LI, F., and LIU, Y. (1999): *The shortest delay algorithm without loss of quality for transmission of stored VBR video*. South Africa Computer Journal, 24: 146-154.

LIU, Y., LEE, C. M., and LI, F. (1999): *Updating scheduling strategy for stored VBR video transmission under limited bandwidth*. Proceedings of 4th International Symposium on Consumer Electronics, Malacca Malaysia, November, 2 (17-19): 31-34.

MCMANUS, J. M., and ROSS, K. W. (1996): *Video on demand over ATM: constant-rate transmission and transport*. Proceedings of IEEE INFOCOM, March, 1537-1362.

MPEG Overview C-Cube Systems

NG, J. K., and SONG, S. (1997): *A video smoothing algorithm for transmitting MPEG video over limited bandwidth*. Proceedings of the Fourth International Workshop on Real-time Computing Systems and Applications (RTCSA), 229 - 236.

PANCHA, P., and ZARKI, M. E. (1993): *Bandwidth requirement of variable bit rate MPEG sources in ATM networks*. Department of Electronic Engineering, University of Pennsylvania, IEEE.

RAO, S., and RAGHAVAN, S. (1999): *Fast techniques for the optimal smoothing of stored video*. Multimedia Systems 7(3): 222-233.

SALEHI, J., ZHANG, Z., KUROSE, J., and TOWSLEY, D. (1996): *Supporting stored video: reducing rate variability and end-to-end resource requirements through optimal smoothing*. Proceedings ACM SIGMETRICS, Philadelphia, PA, May, 222-231.

SIKORA, T. (1997): *MPEG digital video coding standards*. IEEE Signal Processing Magazine, September 14(5).

TSE, C. Y., and LIEW, S. C. (1995): *Video aggregation: an integrated video compression and multiplexing scheme for broadband networks*. IEEE INFOCOM, 439-446.

ZHANG, Z., KUROSE, J., SALEHI, J.and TOWSLEY, D. (1997): *Smoothing, statistical multiplexing and call admission control for stored video*. IEEE Journal on Selected Areas in Communications, August, 15(6).

BIOGRAPHICAL NOTES

Mr. Fei Li gained his B.S. degree in Computer Science from Jilin University, China in 1997. he is now a PhD candidate in the Computer Science Department at Columbia University. his research areas are multi media communication, operating systems and network computing. He has published three international papers and six international conference papers in his research areas. He is an IEEE student member.

Ms Yan Liu is a PhD candidate in the Computer Science Department at Columbia University. She gained her B.E. degree in Radio Engineering in Southeast University, China in 1996. She gained her M.S. degree in Economics from the School of Business in Nanjing University, China in 1998. Her research areas are multimedia retrieval, video segmentation and communication. Ms Yan Liu has published more than twenty papers in her research areas.

MAT: a Mobile Agent System for Supporting Autonomous Mobile Agents

Wei Li

Faculty of Informatics and Communication
Central Queensland University
North Rockhampton, QLD 4702, Australia
w.li@cqu.edu.au

Minjie Zhang

School of Information Technology and Computer Science
University of Wollongong
NSW 2522, Australia
minjie@uow.edu.au

Mobile Agent Template (MAT) is a mobile agent system that is under study and development at the Institute of Computing Technology, Chinese Academy of Sciences and sponsored by the University of Wollongong, Australia. MAT is not an alternative to other mobile agent systems, but is an agent system that can provide the autonomy to mobile agents. MAT tries to support new Web applications, such as the mobile computation, by autonomous and mobile agents. Mobile Thread Programming Model (MTPM), Distributed Task Plan (DTP) and Active State Space (ASS) are integral components on which MAT is constructed. Integration of these three components provides agents with an autonomous work mode and an autonomy-supporting execution environment. In this paper, we define autonomies of agents in the context of mobility and propose our autonomous theories, which are autonomous workflow, asynchronous and localised interactions, and a virtual supporting environment. This paper also outlines current implementation mechanisms of MAT including architecture, program paradigm, distributed task planning and communications. The main contributions of this research are that: (1) workflows are adopted as agents' working modes; (2) a goal-directed and dynamic task planning is used to deal with the heterogeneity and dynamism of networks; and (3) a virtually platform-independent environment is constructed to provide mobile agents with asynchronous, anonymous and fully localised interactions. The innovation of this research is to provide a new solution for novel Web applications such as mobile computations by using MAT.

Keywords: autonomous agent, mobile agent system, distributed supporting environment, thread migration

Copyright© 2001, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the JRPIT copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Australian Computer Society Inc.

Manuscript received: March 2000, revised May 2001
Associate Editor: Hugh Williams

1 INTRODUCTION

It is believed that a new star in the sky of distributed object-oriented systems is the emerging field of mobile agents (Straßer, Baumann and Hohl, 1997). There are many mobile agent systems (Magic, 1998; Gray, 1996; Lange and Oshima, 1998; Object Space) that have been developed since the first prototype, Telescript (White, 1997), was proposed for e-commerce in 1994. There are many ongoing discussions about developments of new mobile agent systems, but few have focused on both mobility and autonomy (Bic, Fukuda and Dillencount, 1996; Cai, Gloor and Nog, 1996). In an application of mobile computation (Gray, *et al*, 1997; Kotz, *et al*, 1997), a user often launches a mobile agent from a laptop that is connected to the Internet, then the user disconnects the laptop from the Internet. The mobile agent travels in the Internet autonomously, retrieving and updating information locally on behalf of its owner. Later, the mobile agent will return to the user's laptop and report the results when the user's laptop is reconnected to the Internet. Mobile agents should have the "intelligence" to perform self-contained navigation and computation, giving mobile agents the powers to adapt to dynamic and heterogeneous networks. This is necessary because in most cases mobile agents cannot interact with their owners. However, it is not easy to define what autonomy is and how to provide mobile agents with autonomies.

This paper focuses on providing agents with autonomies in the context of mobility. In this paper, new concepts, paradigms, and models are proposed and studied as the bases of Mobile Agent Template (MAT) and an agent structure, task description and hosting environment are integral parts to support the autonomies of mobile agents. The contributions of this research are that we adopt the workflow as agents' working mode; we use goal-directed and dynamic task planning to deal with the heterogeneity and dynamism of networks; and we construct a virtually platform-independent environment to provide mobile agents with asynchronous, anonymous and fully localised interactions.

This paper is organised as follows: In Section 2, three kinds of autonomies are defined for clarifying the meaning of autonomy in the context of mobility. In Section 3, the autonomous methodologies, which are Distributed Task Plan (DTP), Active State Space (ASS) and mechanisms for fully localised interactions, are proposed for designing the architecture of MAT. In Section 4, the implementation mechanisms of MAT, including autonomous primitives, DTP planning, and reflections of ASS are addressed. Finally in Section 5, this paper is concluded and directions of future research are given.

2 WHAT ARE AUTONOMIES OF MOBILE AGENTS?

Autonomy is a difficult concept to pin down precisely, but "it means that the system should be able to act without the direct intervention of humans (or other agents), and that it should have control over its own actions and internal state" (Jennings, Sycara, and Wooldridge, 1998). In the context of mobile agents, it is even more difficult to define autonomy. Having investigated some Web applications such as Internet information retrieval and mobile computation for which mobile agents are suitable (Ciancarini, Tolksdorf and Vitali, 1998; Rus, Gray and Kotz, 1997), we identify the necessary features that an autonomous mobile agent should have. By autonomy, a mobile agent is self-contained in navigation, computation and communication when transporting through the underlying heterogeneous and dynamic networks. An autonomous mobile agent should be aware of where to go to find messages or resources, what actions to perform for getting information, and how to deal with exceptions when its performances fail. In the context of autonomy, it should be enough for mobile agents to interact with communication partners only in an asynchronous, anonymous and

localised mode for inter-agent cooperation or resource access. Based on the above features of mobile agents, we have defined the following three types of autonomies in the following subsection.

2.1 Definitions of autonomies

Definition 1: *Mobile autonomy* is the capability of self-navigation of mobile agents through the underlying networks.

The mobile autonomy of mobile agents should be inherent. Mobile agents can be capable of making decisions about their destinations, navigation modes and where to dock when a destination is not reachable. The navigation modes could be serial or in parallel. Normally, a mobile agent completes a distributed task by visiting a series of network nodes serially and processing queries and updating information locally. However, a mobile agent can clone itself and assign a subtask to each of it's duplicates for executing a distributed task concurrently when multiple resources are available and a distributed task can be divided into concurrently executing subtasks. The solutions of subtasks from different mobile agents are combined into a result finally. A mobile agent should also know where to dock temporarily in order to avoid getting lost when a network connection is broken or unavailable, such as the disconnection of a laptop computer from the network. So autonomous mobile agents are dynamically network-aware entities on which Web applications are constructed.

Definition 2: *Computational autonomy* is that a mobile agent can get enough computational functions for accomplishing a distributed task.

A mobile agent should make use of all kinds of computational resources. A mobile agent can invoke functions in different processes or load functions into its process for execution when those functions reside at its visiting network nodes. A mobile agent can also complete a special computational task by executing its own functions carried by the agent when visiting a network node with desirable resources but without necessary computational functions.

Definition 3: *Communicational autonomy* is that a mobile agent can send and receive messages in an asynchronous, anonymous and indirect way.

2.2 Discussion and classification of interactions

However, no matter what degree of autonomy they have, mobile agents need *inter-agent* interactions for cooperation and *agent-hosting execution environment* interactions for resource accesses because any autonomous mobile agent exists in a community, which may be composed of other agents and heterogeneous hosting environments. Besides the interactions with their owners at their home machines, autonomous mobile agents need at least two kinds of interactions during their lifecycles.

1. *Inter-agent interactions* are needed when a Web application consists of more than one mobile agent. Let's consider the following scenario. When a mobile agent realises that its task can be concurrently executed, for example, a mobile agent finds other n interested links in a HTML page while retrieving contents of the page, the mobile agent will duplicate $n-1$ instances with respective retrieval tasks. The agent and its $n-1$ duplicates will transport to n interesting sites for further information retrievals. *Inter-agent* interactions occur when results from different agents are synthesised into a final result for submission to the agent's owner at the home machine.
2. *Agent-hosting execution environment interactions* are needed when a mobile agent accesses resources and services in a local execution environment of network nodes it visits. All the information in a hosting execution environment must be structured for mobile agent accessing, and protocols for retrieving or updating information must be carefully defined.

AUSTRALIAN COMPUTER SOCIETY

INTERNATIONAL COMPUTER DRIVING LICENCE SCHEME



The aim of the scheme is to raise the general level of computer competence, improve productivity in the workplace and reduce user support costs. It is a practical, flexible and cost-effective way of learning, teaching and testing the basic skills required in the everyday use of computers at work, school or in the home.

ICDL is not a "training course", but a competency standard that is acknowledged through testing. ICDL testing will be available through accredited Test Centres, each following a uniform syllabus comprising seven modules.

1. Basic concepts of IT
2. Using the computer and managing files
3. Word processing
4. Spreadsheets
5. Databases/Filing Systems
6. Presentation and Drawing
7. Information network services

A candidate must successfully pass each module before the ICDL is awarded.

The ICDL is positioned to become the most widely recognised, international certification in the field of basic computer use because it targets the full spectrum of the population and it has the support and supervision of the industry's professional societies.

The ACS believes a scheme like ICDL Australia is needed to drive the general level of competence within the Australia workforce and community, and to ensure our nation is equipped to tackle new IT challenges. Additionally, ICDL Australia is an international certification, enabling mobility and recognition for those who undertake the program in many other countries around the world.

National Secretariat
Level 3, 160 Clarence Street
Sydney NSW 2000
Tel: (02) 9299 3666
Fax: (02) 9299 3997
Email: info@acs.org.au
www.acs.org.au

Canberra
Suite 10, 5 Badham Street
Dickson ACT 2602
Tel: (02) 6247 4830
Fax: (02) 6249 6419
Email: mmorgan@acslink.net.au

New South Wales
Level 3, 160 Clarence Street
Sydney NSW 2000
Tel: (02) 9299 2322
Fax: (02) 9299 1280
Email: acsnsw@acslink.net.au

Northern Territory
GPO Box 213
Darwin NT 0801
Tel: (08) 8941 3285
Fax: (08) 8941 3286
Email: acsnt@acslink.aone.net.au

Queensland
Ground Floor Arcade
WorkCover Building
270 Adelaide Street
Brisbane QLD 4000
Tel: (07) 3220 0666
Fax: (07) 3220 0688
Email: mshipqld@acslink.net.au

South Australia
PO Box 2423
Adelaide SA 5001
Tel: (08) 8231 4466
Fax: (08) 8212 3799
Email: vernice@camtech.net.au

Tasmania
Royal Engineers Building
2 Davey Street
Hobart TAS 7000
Tel: (03) 6234 8400
Fax: (03) 6234 2216
Email: acstas@acslink.net.au

Victoria
28 Clarendon Street
South Melbourne VIC 3205
Tel: (03) 9690 8000
Fax: (03) 9690 0201
Email: acsvic@acsvic.com

Western Australia
Suite 4, 342 Albany Highway
Victoria Park WA 6100
Tel: (08) 9470 4878
Fax: (08) 9470 4912
Email: lena@acslink.net.au

If undelivered, please return to:
Australian Computer Society Inc.
P.O. Box Q 534, QVB Post Office
Sydney NSW 1230, Australia

SURFACE MAIL

POSTAGE
PAID
AUSTRALIA

Information Age
AUSTRALIA'S IT NEWS TRENDS MAGAZINE

Print Post Approved: 255003/00967

Journal of Research and Practice in Information Technology

Print Post Approved: 255003/01660

1047465
MR A.W. MCMEEKIN MACS
EXECUTIVE DIRECTOR
INFORMATION TECHNOLOGY SERVICES
MONASH UNIVERSITY
CLAYTON VIC 3168

CHANGE OF ADDRESS

ACS MEMBERS:
Please send all address changes to the ACS
Phone: +61 2 9299 3666 Fax: +61 2 9299 3997
Email: membership@acslink.net.au

NON ACS MEMBERS:
Please make any changes on this sheet and fax it to
IDG Communications PTY LTD on +61 2 9439 5512

This is my ☐ Home address ☐ Work address

Mr/Mrs/Miss/Ms..... Family Name.....

Job Title

Company

Address

.....

.....

Telephone (H)

Telephone (B)

Email Address

There are new features in interactions among autonomous mobile agents or between autonomous mobile agents and hosting execution environments. In the context of autonomy, interactions should be fully asynchronous, anonymous and localised between communication partners. Firstly, the asynchronism does not require a strict agreement on communication time while the synchronism implies that any interaction between communication partners depends on complex protocols and synchronous mechanisms. The code load required for mobile agents to deal with their protocol, and synchronous mechanism is too high. Mobile agents would have too many code loads for the protocol and the synchronous mechanism accomplishments. Secondly, if an autonomous mobile agent must know the communication partners' name for interaction, naming services would be involved in agent communications. The naming services result in extra network (not local) interactions between mobile agents and naming service agencies. Too many interactions might weaken the autonomy we pursue in many Web applications. Also, it is not necessary for an autonomous mobile agent to know other agents' names because it is enough for an agent to interact with others as long as it can identify that its communication partners and itself belong to the same application. Finally, any interaction should be localised. An important advantage of mobile agents over other distributed technologies is the locality of distributed information processing. Locality leads to advantages in terms of high efficiency, reliability and flexibility. So interactions between autonomous mobile agents and their communication partners should not introduce any extra network communication.

In the concept of wireless networks (low band) and mobile computation (partially connected devices), minimisation of network communications and localisation of interactions are the desirable goals of MAT. To achieve those aims, we cannot use the traditional communication models, such as the direct communication model and the rendezvous model (Acharya, Ranganathan and Saltz, 1996; Baumann, Hohl, and Radouniklis, 1997; Gray 1996), or the meeting-oriented model (Magic, 1998; Peine and Stolpmann, 1997). Instead, we design a new model: *ASS (Active State Space)*, which is a third party to facilitate interactions between autonomous mobile agents and their communication partners for cooperation or resource accesses.

3 HOW TO SUPPORT AUTONOMIES: METHODOLOGY OF MAT

In this section, we demonstrate how to introduce desirable autonomies into mobile agents by using the virtual autonomy-supporting environment and the autonomy-supporting work mode.

3.1 Abstract description to heterogeneous execution environments in MAT

Mobile agents exist in heterogeneous networks such as the Internet. Resources are formatted and managed with different modes. Accessing different system resources directly by mobile agents results in the following limitations:

1. Increasing complexity of the application-level programming. It is evident that programmers must write different code for accessing different type or the same type of resources. For example, an *html* file is described by path */user/wli/document/index.html* in UNIX and described by path *c:\wli\document\index.html* in Windows. In order to access resources in heterogeneous networks, programmers should be familiar with different networks and operating systems.
2. Increasing code loads carried by mobile agents. The more heterogeneous the network that a mobile agent travels in, the heavier the code load that a mobile agent carries. A mobile agent with a too heavy code load is not a compact entity suitable for Web applications in the context of minimising network communications to deal with low band and unreliable networks such as the wireless network.

3. Decreasing reusability of agent codes. Any effort to extend a mobile agent-based application will result in the modification and addition of agent code.

In MAT, we use Active State Space (ASS) to provide an integrated description of heterogeneous resources that are accessed by mobile agents. ASS abstracts resources of different systems into data integrity so as to deal with the heterogeneity and the dynamism of the hosting execution environments. Mobile agents use virtual resources to access the real resources. ASS is responsible for mapping between abstract resources and real resources. ASS provides mobile agents with platform-independent resource accesses. In this point, ASS is quite similar to JVM, which provides platform-independent Java instructions to programmers. Another motivation for use of virtual resources is to provide a secure interface between mobile agents and system resources. It is necessary for ASS to perform security checks on some sensitive operations such as deleting files because a malicious or badly programmed agent may possibly contain dangerous code. Comparison of direct resource access to virtual resource access is graphically illustrated in Figure 1.

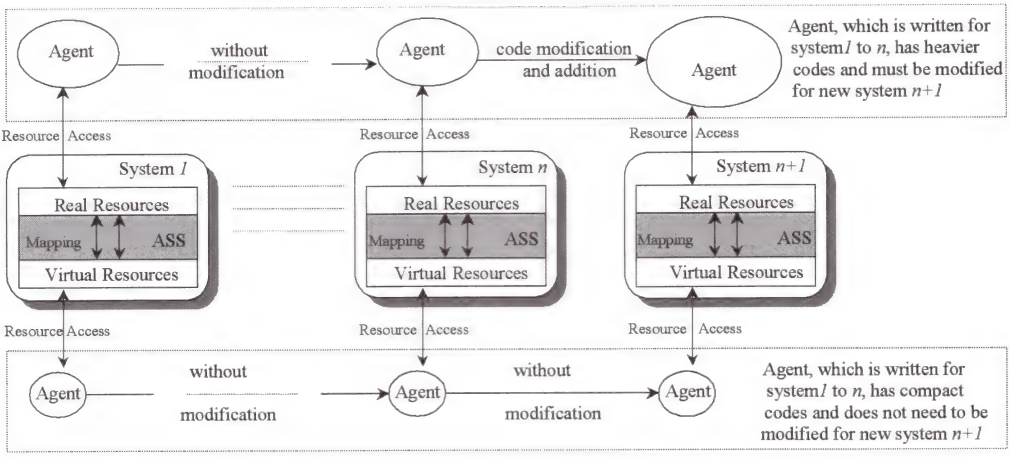


Figure 1: Comparison of direct resource access to virtual resource access

3.2 Asynchronous and localised interactions between communication partners in MAT

Asynchronous, anonymous and localised interactions are main features that must be satisfied in the context of Web applications based on autonomous mobile agents. In MAT, we use Resource Space, a component of ASS, to coordinate actions of mobile agents for cooperation in a task accomplishment. We use the following scenario to demonstrate the asynchronous, anonymous and localised coordination model for *inter-agent* interactions.

When a mobile agent *PA* knows that its distributed task DT can be executed concurrently on destinations D_1, D_2, ..., D_{n-1}, and D_n., the agent PA (called Parent) duplicates *n-1* instances SA_1, SA_2, ..., and SA_{n-1} with respective subtasks ST_1, ST_2, ..., and ST_{n-1}. Each duplicate will transport to D_1, D_2, ..., and D_{n-1} respectively, and finally, the parent agent *PA* will transport to the destination D_n with subtask ST_n. Subtasks ST_1, ST_2, ..., ST_{n-1} and ST_n can be the same or different. If our aim is to exploit distributed resources for executing distributed tasks for fast response, the subtasks should be same, such as, getting the same program table from different TV web sites. If our aim is to exploit distributed resources for retrieving different information, the subtasks should be different, such as, getting different price table of the same travel from different travel agencies. Because SA_1, SA_2, ..., SA_{n-1}, and *PA* concurrently execute subtasks ST_1, ST_2,

....., ST_{n-1}, and ST_n divided from DT, the results from different mobile agents need to be combined into a final result at a network node Site on which agents agree. Results of subtasks ST_1, ST_2, ST_{n-1}, and ST_n from different agents can be described by the following Atomic Record type of Active State Space.

SAR=<String Results, Integer ApplicationID, Integer TaskID>

If ST_1, ST_2, ..., ST_{n-1} and ST_n are the same, agents SA_1, SA_2, ..., SA_{n-1}, and *PA* will compete for being alive at the same network node Site. The agent, which arrives at *Site* first, is regarded as the first one to accomplish this special subtask. This agent will be continuously alive, and others will exit their execution. It can be identified by the following Atomic Record as the Label in Resource Space of ASS whether an agent just transporting to *Site* is the first one to accomplish the special task.

LAR=<String State, Integer ApplicationID, Integer TaskID>

If a mobile agent cannot find the LAR in the current ASS, it is the first one to accomplish the special task identified by ApplicationID and TaskID, then it will put a LAR into the ASS. If a mobile agent finds the LAR in ASS, it is not the first one to accomplish the special task and will exit its execution.

If ST_1, ST_2, ..., ST_{n-1} and ST_n are different, all the results of subtasks ST_1, ST_2, ..., ST_{n-1}, and ST_n from different agents SA_1, SA_2, ..., SA_{n-1}, and *PA* need to be combined into a final result. The parent agent *PA* is responsible for the solution synthesis. The duplicated agents SA_1, SA_2, ..., and SA_{n-1} will exit their executions after putting their results into the current Resource Space of ASS at *Site*. The *inter-agent* interactions of the above scenario are graphically illustrated in Figure 2.

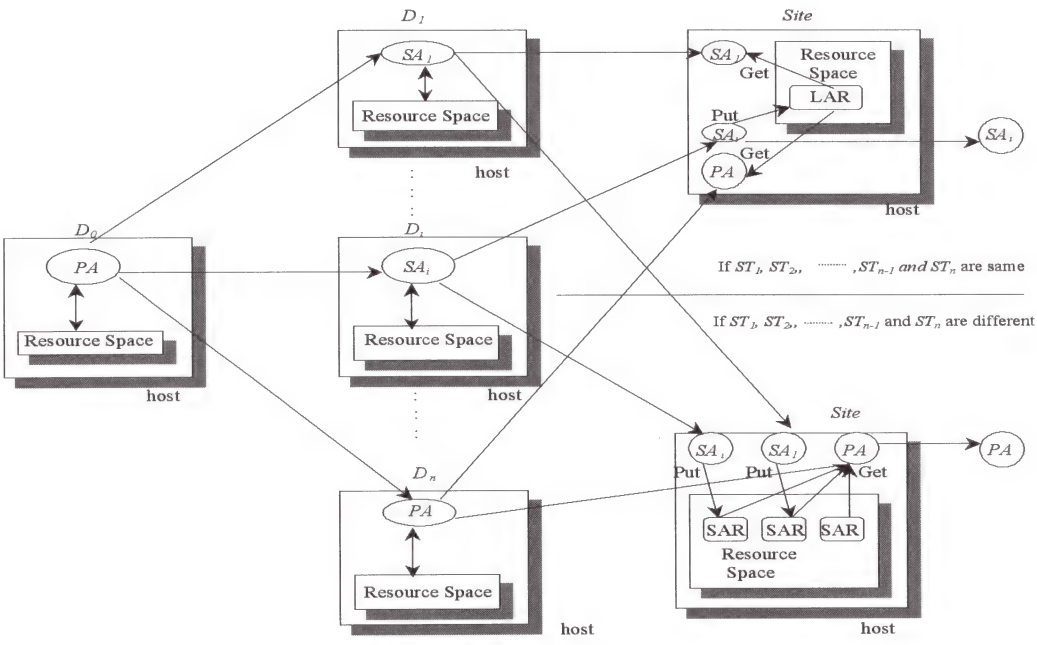


Figure 2: Inter-agent interactions by Resource Space of ASS

3.3 Working mode of workflow for mobile agents in MAT

An autonomous mobile agent should be able to exhibit goal-directed behaviours, which depends on a dynamically planned, continuous and autonomous workflow. In order to provide mobile agents with an autonomous workflow, we must solve two problems:

1. Maintaining the persistence of an agent's thread. This means the whole mapping (including the execution point of the thread) of an agent object must be restored after the agent migration. The weak mobility model, which is widely used in many mobile agent systems, such as Odyssey (Magic, 1998), Voyager (Object Space), Java-To-Go (Li and Messerschmitt, 1996), Aglets (Lange and Oshmina, 1998), Facile (Thomsen, Leth and Prasad, 1993), Tocoma (Johansen, Reuese and Schneider, 1995), Mole (Object Space) and Grasshopper (IVK, 1999), is not suitable for supporting the persistence of agents' threads. An agent's execution can only start from the beginning or from a special method rather than the stopped point after agent migration, so the persistence of an agent's thread cannot be maintained by weak mobility models. On the other hand, the strong mobility supports the persistence of agents' threads by transparently capturing the mapping of an agent's thread when an agent wants to migrate, transporting the captured mapping and restoring the transported mapping after the agent migration. But the strong mobility is often constructed on special languages. Systems based on special or specification-modified languages, such as Telescript (White, 1997), Agent Tcl (Gray, 1996), Ara (Peine and Stolpmann, 1997) and Sumatra (Acharya, Ranganathan and Saltz, 1996), are accepted by multiple platforms with difficulty. An evident example is that General Magic rewrites its mobile agent system, Telescript, into Odyssey by using Java in order to be widely accepted by heterogeneous platforms.
2. Permitting the modification of its behaviours when a mobile agent transports under the computational networks. This means a mobile agent cannot only execute a predefined workflow, but also modify or reconstruct its workflow in order to deal with the heterogeneity and dynamism of networks. Behaviours of an agent are exhibited by its code. Dynamic reconstruction of a workflow means that the agent kernel and its behaviours, which are defined by a user's objectives, should be loosely coupled and cannot be programmed in the same program unit because an agent needs to modify its code for adapting to dynamic networks.

Limitations of current technologies of agent migration necessitate the design of a new agent system model, MAT, for supporting autonomous mobile agents. In MAT, we program mobile agents with Mobile Thread Programming Model (MTPM). Contrary to many strong mobility technologies, MTPM is an application-level rather than a system-level model, that simulates the persistence of threads after agent migrations. An advantage of MTPM is that it is highly efficient. MTPM can simulate the persistence of an agent's thread without capturing, transporting and restoring the thread stacks and heaps with huge system-dependent information. MTPM does not need any modification to JVM specifications, and it uses two new mechanisms, Serialisation and Remote Method Invocation (RMI), provided by Java 1.1 and later releases, to provide persistence to an agent's thread. MTPM is outlined in Subsection 4.2. When it is created, a mobile agent plans its Distributed Task Plan (DTP) that is the implementation model of MTPM. An execution of a DTP generates a continuous and autonomous workflow for a mobile agent. When executing a DTP, a mobile agent can transport in networks, retrieve resources at remote hosts and asynchronously interact with other agents. At any time when current DTP fails, a mobile agent can replan a new DTP according to dynamic networks for an alternative accomplishment of a user's objectives. A DTP is a reusable template that is a static description of a distributed task for the execution by a mobile agent. An

execution of a DTP is an autonomous workflow for the agent. Whenever a mobile agent needs to modify its behaviours for adoption of the dynamism of networks, it can reach the goal by replanning its DTP. DTP is outlined in Subsection 4.2 and detailed in (Li and Zhang, 1999). The work mode of mobile agents in MAT is graphically illustrated in Figure 3.

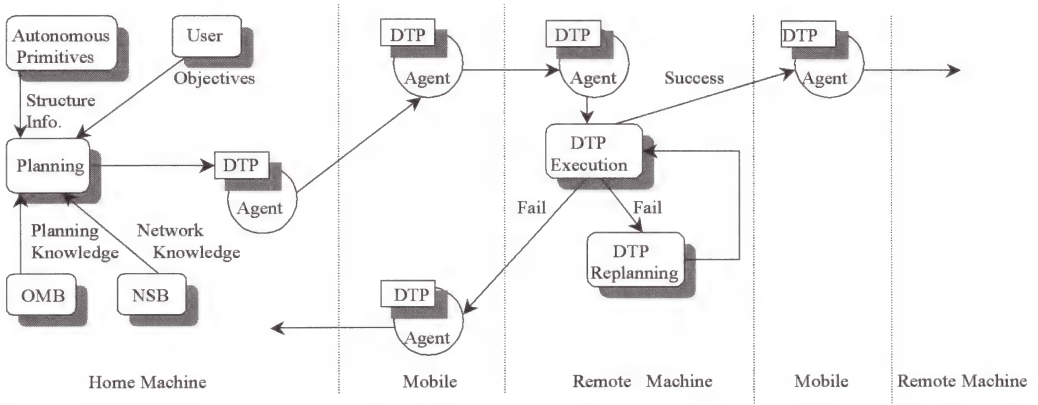


Figure 3: The planning, execution and replaning of DTP

4 IMPLEMENTING AUTONOMIES: MECHANISM OF MAT

4.1 ASS: distributed supporting environment

There are three motivations for the design of ASS. The first one is to provide communication facilities for autonomous mobile agents. Interactions among agents, by means of ASS, are not involved in any network communication and naming service. Interactions among agents are fully asynchronous, anonymous and localised. That means a mobile agent can put messages (called *Record* objects) in ASS, which another agent can get later. The second one is to provide a uniform description of heterogeneous resources that are accessed by mobile agents. ASS abstracts the resources of different systems into the data integrity so as to deal with heterogeneity and dynamism of the hosting execution environments. Mobile agents use virtual resources for accessing the real resources. ASS is responsible for mapping between abstract resources and real resources. The last motivation is to provide a secure interface between mobile agents and system resources. It is necessary for ASS to perform security checks on some sensitive operations such as deleting files because a malicious or bad-programmed agent possibly contains dangerous code.

4.1.1 Architecture of ASS

ASS is composed of *Resource Space*, *Security Space* and *Operation Space*. Virtual resources or messages for *inter-agent* communications are stored in the *Resource Space*. Any requirement to access *Resource Space* submitted to ASS by a mobile agent will fire an *Operation* object in the *Operation Space*. For the acceptance or rejection of a mobile agent's requirements, the *Operation* object needs to match the requirements with *Security Rules* in *Security Space*. This interaction procedure between mobile agents and hosting execution environments is called *Reflection of the Operation Object* in ASS. So every *Operation* object has a segment of *Reflection* codes. A successful reflection will change the states of ASS, and a failed reflection yields no influence on the ASS.

In an ASS, any component, such as a virtual resource, a security rule or an operation, can be installed and deleted dynamically by the ASS itself or mobile agents. The components, installed by

the ASS itself, are the common resources or protocols for all the agents and usually exist for a long period. The components, installed by agents, are the special resources or protocols for an application and usually exist for a short period. Components in the above three spaces are integrated into abstract description called an Active Record as defined below.

Definition 4: A *Record* in an ASS is composed of objects that are called *sub-records*, i.e. *<Record>::=<{subrec,}*>*, where subrec denotes any *sub-records* which compose the *Record*.

Definition 5: A *Record* is an *Atomic Record* **if** its every *sub-record is* not a *Record* object.

Definition 6: A *Record* is a *Coupled Record* if at least one of its *sub-records* is a *Record* object.

A file in a local file system can be structured with an *Atomic Record* object with the type of *<String path, String Type, Date UpdateTime, Username owner>*. Then the *Atomic Record* object of *<'/user/wli/document/', 'html', 2/22/99, wli@cowan.edu.au>* represents system file resources: all the *html* files that are located in the directory of *user/wli/document/*, owned by user *wli@cowan.edu.au* and updated in *February 22, 1999*.

A message, which a mobile agent puts in an ASS for communications with other agents, can be structured with an *Atomic Record* with the type of *<String Results, Integer ApplicationID, Integer TaskID>*. Every application has a unique identification *ApplicationID*, and every distributed task of the same application has a unique identification *TaskID*. Any agent can get the results of a distributed task by providing the *ApplicationID* and *TaskID* to the mechanism of the resource matching of the ASS.

A security rule of ASS can be structured with a *Coupled Record* object with the type of *<Integer AgentID, String Operation, AtomicRecord Rec>*. Then the *Coupled Record* object of *<*, 'r', Rec_1>* represents that any agent can read *Record* Rec_1. Another *Coupled Record* object of *<agentid, 'rdu', Rec_1>* represents that mobile agent identified by *agentid* can read, delete and update *Record* Rec_1.

Each Application ID, Task ID or Agent ID is 128 bits, and we use a special algorithm to generate IDs. Our ID algorithm is very similar to the algorithm of Microsoft for generating COM GUID (Microsoft). Microsoft demonstrated that the algorithm could generate different IDs in 3240 years at the generating speed of 1000000 IDs per second.

An *Operation* object of an ASS can be structured with an *Atomic Record* object with the type of *<String OperationName, Operation OpObj, Username Owner>*. Where the *OperationName* is the name of the *Operation*, *OpObj* is the instance of the *Operation* and the *Owner* is the entity (agent or local system) to install the *Operation*. Every *Operation* object has a *reflection* method. The *reflection* method is executed when the *Operation* object is called by ASS.

A retrieval requirement from a mobile agent to a *hosting execution environment* can be structured with a *Coupled Record* object with the type of *<Integer AgentID, String Operation, Boolean Synchronisation, AtomicRecord Rec>*. Then the *Coupled Record* object of *<agentid, 'r', true, <'/user/wli/document/', 'html', *, *>>* represents that a mobile agent, identified by *agentid*, wants to read all the *html* files in the directory of */user/wli/document/* in a *synchronous* mode. If the resource matching and the security inspection succeed, the ASS will return a vector of references to the required *html* files.

4.1.2 The reflection mechanism of ASS

ASS has three Object Spaces, which are *Resource Space*, *Security Space*, and *Operation Space*. The *Resource Space* stores *Atomic Records*. These records abstractly describe system resources of the *hosting execution environments* or messages that mobile agents put for asynchronous interactions.

The *Security Space* stores *Coupled Records*. These records describe security rules that determine which agents can do what kinds of operations on which records in the *Resource Space*. The *Operation Space* stores *Atomic Records*. These records describe *Operation* objects that are identified by names and are called by the ASS when mobile agents interact with the ASS. *Operation* objects can be installed by an ASS system or by mobile agents for some special reflection operations to interaction events.

Every *Agent Server* hosts an ASS object. When a mobile agent transports into a network node, the agent gets the reference to an ASS object from the *Agent Server*. An important feature of an ASS is that *Operation Records*, which are responding entities to interaction events, can be programmed, and can be installed dynamically by agents or an ASS itself. The whole procedure of an interaction event from the submission of an interaction requirement to the result return is graphically illustrated in Figure 4.

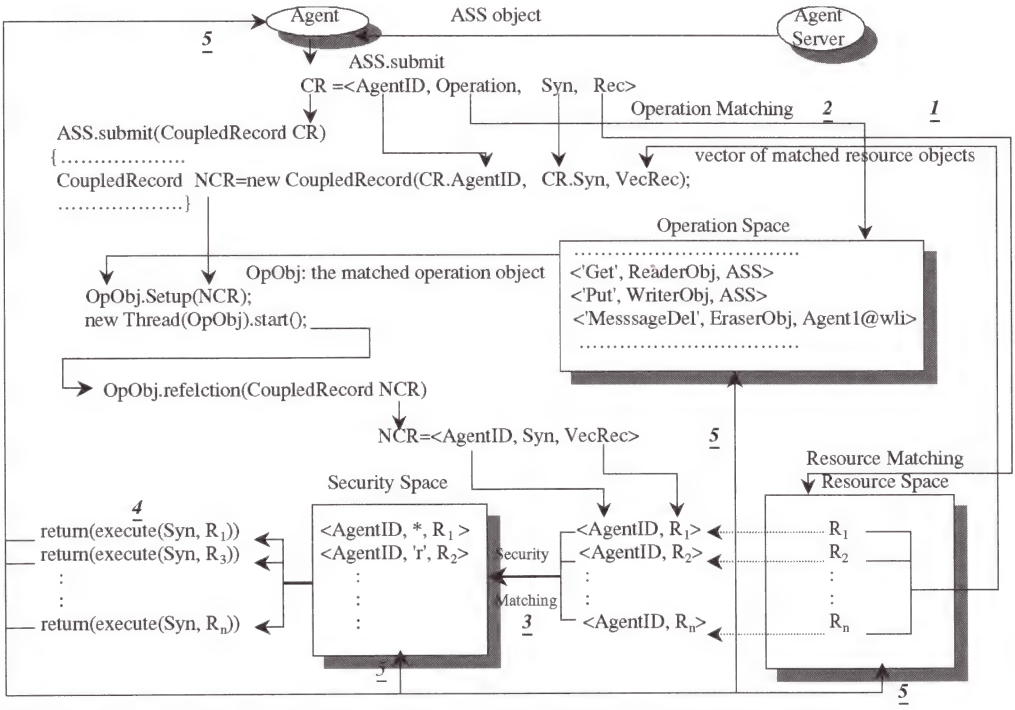


Figure 4: The reflection procedure of ASS object to an interaction requirement from a mobile agent

Every interaction requirement can be represented by a *Coupled Record*: *<AgentID, Operation, Syn, Rec>*, which means: a mobile agent identified by *AgentID* hopes to execute an *Operation* on all the *Records* matching the template record *Rec* in a synchronous mode *Syn*. The agent provides the above requirement to an *ASS* object with the *submit* method of the *ASS* object to which the agent has a reference. The reflection procedure of an ASS to the interaction requirement goes through the following steps:

1. The ASS object searches the whole *Resource Space* to identify records that match *Rec*, and uses these records to construct a vector *VecRec*.
2. The ASS object searches the whole *Operation Space* to identify a record that matches

Operation; generating a thread for the matched operation object *OpObj*, and starting the execution of the thread.

3. The *Operation* thread constructs a record *SR=<AgentID, R>* for each record *R* in vector *VecRec* and searches *Security Space* to identify the presence of a record that matches *SR*.
4. If a matched record is found in *Security Space*, *OpObj* will execute the required *Operation* on *R* in the synchronous mode *Syn*.
5. The results of the *Operation* can be returned to the agent or stored into *Resource Space*, *Security Space* or *Operation Space*.

Some ideas for the *ASS* design come from Jada (Ciancarini and Rossi, 1997). In Jada's space, objects are passive. *ASS* takes Jada's concepts one step further by allowing components of ASS to actively respond to interaction events and by allowing components to be installed, updated and deleted dynamically by the *ASS* itself or mobile agents.

4.2 Workflow template DTP for generating a continuous and autonomous workflow

In this subsection, we introduce the Distributed Task Plan (DTP), which is a flexible implementation model of MTPM for generating continuous and autonomous workflows for mobile agents. MTPM simulates the state persistence of an agent's thread by Serialisation and RMI without introducing any new spatial and time complexity. DTP is a workflow template that uses the programming paradigm of MTPM (Zhang and Li, 1999).

In order to obtain desirable autonomies for mobile agents, we firstly provide enough programming components and autonomous primitives, which are used to construct a DTP and further to provide mobile agents with autonomies in the navigation and the computation. In MAT, four kinds of autonomous primitives are defined for DTP designing.

1. Mobile primitives define the mobility of an agent. A mobile agent can merely transport itself to the next destination from the current network node by calling the *simple migration primitive*, or clone and transport each of its duplicates to different destinations by calling the *multiple migration primitive*.
2. Computational primitives define invocations of computational resources. A computational primitive specifies where to find the current computational procedure, how to load it and how to execute it. By using the computational primitives, a mobile agent realises that (a) the current computational procedure is carried by the agent or is resident at the visiting node; (b) the procedure should be started in a different process or loaded into its own process; and (c) how to run the computational procedure, e.g. synchronously or asynchronously.
3. Solution synthesis primitives define the combination of multiple solutions from different mobile agents. The solution synthesis is needed when a task is divided into several subtasks and executed by different mobile agents concurrently. It is highly efficient to divide a task into several subtasks and assign these subtasks to different mobile agents for execution when the task can be executed concurrently and multiple resources are available.
4. Control primitives define the execution flow of mobile primitives, computational primitives and solution synthesis primitives. Enough control structures of control primitives are needed to efficiently coordinate the executions of all the above three primitives.

Having defined primitives, we provide a reasonable model: DTP, which is used to depict distributed tasks for mobile agents by advantages of those pre-defined autonomous primitives. Normally, DTP is composed of all the four kinds of autonomous primitives.

A DTP consists of primitives, which are arranged into two lists. A list, called a control queue (CQ), only contains control primitives, and the other list, called a reusable primitive list (RPL),

contains any primitives except control primitives. The architecture of DTP is graphically illustrated in Figure 5 (concrete meanings of primitives of Figure 5 are defined in (Li and Zhang, 1999)).

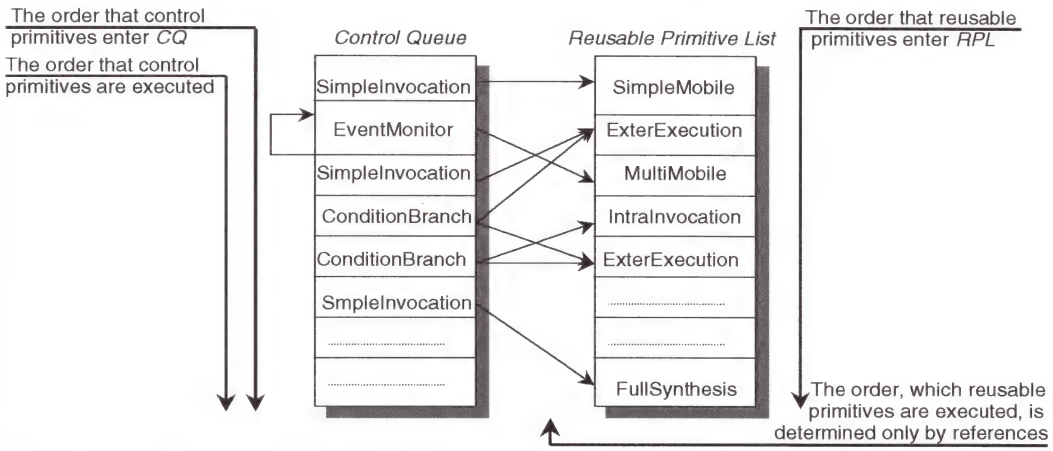


Figure 5: The architecture of DTP

When being created, a mobile agent plans its own DTP for the execution of a distributed task satisfying user's requirements. The planning procedure includes Objective Matching, Primitive Selection and DTP Generation by using users' requirements, network state information and task features. A mobile agent can also replan its DTP when the current DTP fails during execution. The planning of DTP is detailed in the next subsection.

A mobile agent has a reference to its DTP, and a DTP has a reference to a CQ. All the objects, from the agent itself, DTP to autonomous primitives, have a run() method. Every objects' run() method just calls the run() method of another object to which the former has a reference. The autonomous workflow of a mobile agent is generated when the mobile agent executes its DTP by running its run() method as shown in Figure 6.

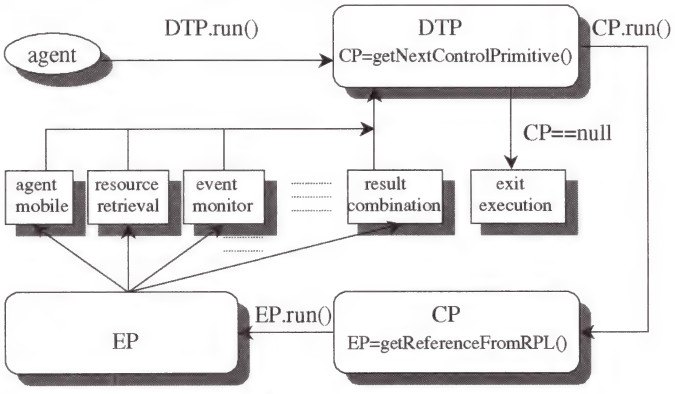


Figure 6: The continuous and autonomous workflow of agent by DTP

The order of primitives in a CQ is important. The control primitives in a CQ are executed sequentially. A control primitive in a CQ has one or more references to primitives in a RPL corresponding to which type the control primitive is. A reference to a control primitive in a CQ depicts a possible invocation to a primitive in a RPL. The order of primitives in a RPL is not

important because the invocations to them are determined only by references of control primitives in a CQ (see Figure 5). A RPL is just a repository of autonomous primitives that a mobile agent may need to execute when transporting in the underlying networks. So a mobile agent only executes control primitives in a CQ one by one, then further executes primitives in a RPL. An execution of a DTP is a continuous and autonomous workflow of a mobile agent. Constructing a complex workflow by using DTP provides a mobile agent with autonomy, flexibility and reusability in distributed applications.

4.3 Planning and replaning of DTP

According to a user's objectives, a DTP object should be planned either by a mobile agent or a static agent. In order to make mobile agents lightweight in code load and to support complex planning strategies, we move the planning algorithm of DTP (Li and Zhang, 1999) from the mobile agent in a previous version of MAT into the static agent (called *Planner*) into our current version. When a mobile agent is created, it gets a task from its owner. The task is described by the following *Atomic Record.*

OAR=<String TaskLable, String Objectives, Integer ApplicationID, Integer TaskID>

With following codes, the agent puts its *OAR* into a current ASS, and later gets its DTP object from the current ASS. The received DTP object is matched to the user's objectives described by *OAR*. Once a mobile agent gets its DTP object, the agent executes it.

```
ASS.submit(new CoupledRecord(this.AgentID, 'Put', true, OAR));
AtomicRecord   DTPID=new(*, OAR.Objectives, OAR.ApplicationID, OAR.TsakID);
AtomicRecord   DTPAR;
while ((DTPAR= ASS.submit(new CoupledRecord(this.AgentID, 'Get', true, DTPID)))==null)
      { Thread.sleep(time); };
DTP   DTPObject=DTPAR.DTPObject;
DTPObject.run();
```

When the agent fails to execute its current DTP object due to the dynamism of networks, it also uses the above codes to get a new DTP object, which is replanned by a Planner at its visiting network nodes, from the current ASS. The new DTP is an alternative workflow for the mobile agent to accomplish its owner's objectives in dynamic networks.

There is one Planner running on every distributed ASS. A Planner continuously gets users' objectives (described by *OAR*) from the current ASS, generates DTP objects and also puts them into the ASS using the following codes.

```
AtomicRecord   OARTemplate=new('task', *, *, *);
  while (true) {
    AtomicRecord   OARObject=ASS.submit(new CoupledRecord(this.AgentID, 'Get', true, OARTemplate));
    if (OARObject!=null) {
      DTP   DTPObject=planning(OARObject.Objectives);
      AtomicRecord   DTPAR=new AtomicRecord(DTPObject,OAR.Objectives, OAR.ApplicationID, OAR.TaskID);
      ASS.submit(new CoupledRecord(this.AgentID, 'Put', true, DTPAR)); }
else Thread.sleep(time); }
```

The procedure of DTP planning, running and replanning is graphically illustrated in Fig. 7.

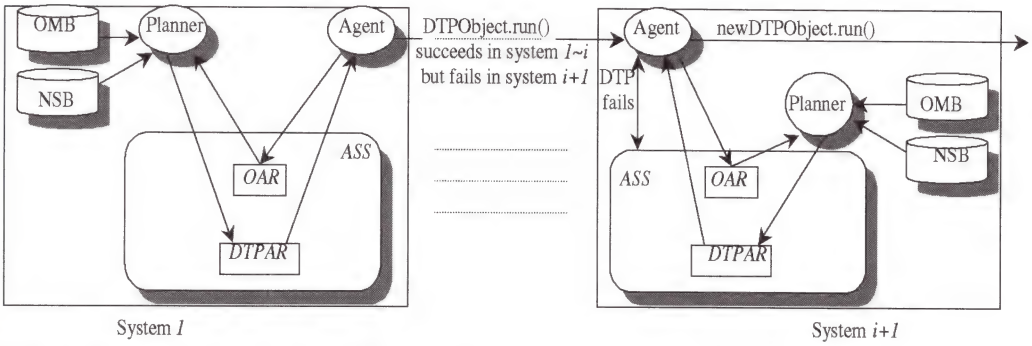


Figure 7: Planning, running and replanning of DTP

The Planner matches a user's objectives with entries in Objective Matching Base (OMB) to decide which primitives are used. The selected primitives are arranged into a CQ or a RPL according to their types. The references of primitives in the CQ to primitives in the RPL are set up at the same time. The developers of MAT applications define the OMB. An OMB consists of reusable entries. The matching method uses the algorithm that we prefer to call *Requirement Propagation* to match a user's objectives with pre-defined entries, which determine DTP planning. The Planner also retrieves a Network State Base (NSB) for information about current networks. A NSB is periodically updated by a MAT information server in order to reflect the dynamic networks. A concrete example of DTP planning is graphically illustrated in Figure 8. In this example a user wants to get a file *ProgramTable* from remote network node *butterfly* when the file is updated after *February 22, 1999.*

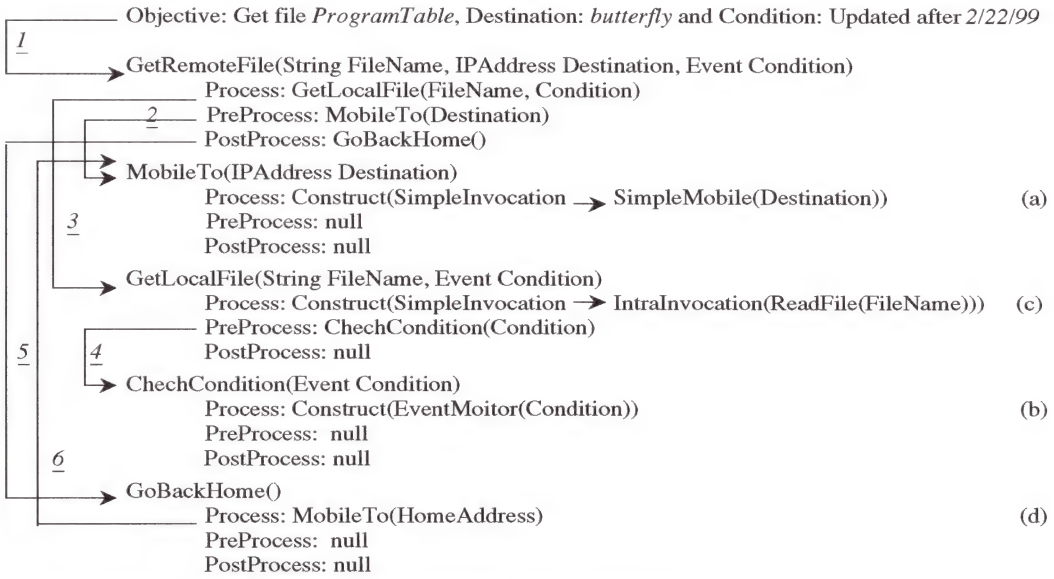


Figure 8: DTP planing: objectives matching with reusable entries in OMB

Each entry of OMB has four components, an entry name with parameters, *Process*, *PreProcess* and *PostProcess*. Process is executed after the execution of *PreProcess* and before the execution of *PostProcess*. The executions of *Process*, *PreProcess* or *PostProcess* can possibly access other entries, so the objective matching is the Requirement Propagation by reusing entries in OMB. In the above example, the objective matching is in the order of 1-2-3-4-5-6 and the DTP planning is in the order of (a)-(b)-(c)-(d). The workflow generated by executing the planned DTP will be:

Step 1: Transporting to *butterfly*,
Step 2: Monitoring file *ProgramTable* until it is updated after *February 22, 1999*,
Step 3: Reading the file, and
Step 4: Taking the file back the home machine.

The replanning of DTP is the same as the planing of DTP but the Planner uses the OMB and the NSB in the current node rather than those in the home machine of a mobile agent. The replanning of DTP gives a mobile agent an alternative workflow to accomplish its owner's objectives when current DTP fails.

5 CONCLUSION

Partially connected computers, such as laptops, personal digital assistants or modem-connected home computers, are involved with the development of the network, especially the wireless network. Applications launched from them are usually called mobile computation, which depends on the autonomies of mobile agents. MAT is a mobile agent system, which is embedded with high autonomies. As we know, many mobile agent systems, such as Telescript, Voyager and Aglets, do not or do not highly support autonomy of agents because their goal is not to support the novel Web application of mobile computation. Only the mobile agent system Agent Tcl pays some attentions to mobile computation. But none of them proposed the autonomy theory for the mobile agents. However, mobile computation will become a popular Web application with the development of wireless networks and more and more mobile devices, such as laptop computers and mobile phones. The feature, which distinguishes MAT from other agent systems, is that MAT concentrates on autonomies in the context of mobile agents. For supporting both mobility and autonomy, many concepts, paradigms and models, have been proposed for MAT implementation, which mainly include the autonomies of mobile agents, MTPM, DTP and ASS. All the proposed models are implemented currently. In MAT, a mobile agent is an autonomous entity that can plan its workflow and roam in networks for resource accessing or cooperation with other agents in asynchronous, anonymous and localized modes. By using MAT, a developer can use all kinds of templates, including DTP, Active Record and Object Matching Rule, to construct Web applications. A user of MAT applications only needs to provide his objectives and other necessary information such as features of tasks (concurrent or serial). Mobile agents can determine goal-directed work and cooperation modes according to users' requirements, and try alternatives when an effort fails.

MAT is a new prototype system implemented in Java that concentrates on providing agents with both mobility and autonomy. This paper summed up its current methodologies and implementation. A web application titled Intelligent Retrieval for the Information on the Web was constructed on MAT. This application supports the use of mobile devices. Another application titled Distributed Swarm is under construction. This application enables the Swarm (Swarm Design Group) into a distributed system. Our future work will focus on investigating the suitability and the effectiveness of MAT in Web applications, such as Internet information retrieval, electronic commerce and Computer Support Cooperative Work (CSCW). From feedback of the investigations, we may find problems in MAT, and make improvement to the methodology, model and their implementations.

ACKNOWLEDGMENTS

Authors would like to thank Mrs Anne Fuller for her proof reading of the paper. Authors cheerfully acknowledge the anonymous reviewers for their criticisms, comments and suggestions, which have been very helpful in improving the quality of this paper.

REFERENCES

ACHARYA, A., RANGANATHAN, M., and SALTZ, J. (1996): Sumatra: A language for resource-aware mobile programs. *Mobile Object System: Towards the Programmable Internet, Lecture Notes in Computer Science,* 1222:111-130, Berlin, Springer-Verlag.

BIC, F., FUKUDA, M. and DILLENCOURT, M. B. (1996): Distributed computing using autonomous objects. *IEEE Computer.*

BAUMANN, J., HOHL, F. and RADOUNIKLIS, N. (1997): Communication concepts for mobile agents. *Proc. of the First International Workshop on Mobile Agents, Lecture Notes in Computer Science,* 1219:123-135, Berlin, Springer-Verlag.

CAI, T, GLOOR, P. and NOG, S. (1996): Dartflow: A workflow management system on the web using transportable agents. *Technical Report TR96-283*, Department of Computer Science, Dartmouth College, Hanover.

CIANCARINI, P. and ROSSI, V. (1997): Jada: Coordination and communication for Java agents. *Mobile Object System, Lecture Notes in Computer Science*, 1222:213-226, Berlin, Springer-Verlag.

CIANCARINI, P., TOLKSDORF, R., and VITALI, F. (1998): Coordinating multiagent applications on the WWW: A reference architecture. *IEEE Transactions on Software Engineering*, 24(5):362-375.

MAGIC G. (1998): Introduction to the Odyssey API. Available at http://www.generalmagic.com/agents/ odysseyIntro.pdf.

GRAY G. (1996): Agent Tcl: A flexible and secure mobile-agent system. *Proc. of Fourth Annual Tcl/Tk Workshop*, Monterey, California.

GRAY, G., KOTZ, D., NOG, S., RUS, D. and CYBENKO, G. (1997): Mobile agents for mobile computing. *Proc. of the Second Aizu International Symposium on Parallel Algorithms/Architectures Synthesis*, Fukushima, Japan.

IKV (1999): Grasshopper. Available at http://www.ikv.de/products/grasshopper.html.

JENNINGS, N. J., SYCARA, K. and WOOLDRIDGE, M. (1998): A Roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems,* 1(1):7-38, Kluwer Academic Publishers.

JOHANSEN, D., RENESSE, R. and SCHNEIDER, F. B. (1995): An introduction to the TACOMA distributed system. Computer Science Technical Report 95-23, University of Tromso, Norway.

KOTZ, D., GRAY, R., NOG, S., RUS, D., CHAWLA, S., and CYBENKO, G. (1997): Agent TCL: Targeting the needs of mobile computers. *IEEE Internet Computing*, **1**(4):58-67.

LANGE, D. B. and OSHIMA, M. (1998): Programming and developing Java mobile agents with Aglets. Forthcoming Booking, Addson-Wesley.

LI, W. and ZHANG, M. (1999): Distributed task plan: A model for designing autonomous mobile agents. *Proc. of International Conference on Artificial Intelligence*, Las Vegas, 336-342.

LI, W., and MESSERSCHMITT, D. G. (1996): Java-To-Go. Technical report, Dept. of EECS, University of California, Berkeley, available http://ptolemy.eecs.berkeley.edu/dgm/javatools/java-to-go/.

Microsoft, COM Specification. Available at http://www.microsoft.com/com/resources/comdocs.asp

Object Space, Voyager Core Technology 2.0 User Guide. Available at http://www.objectspace.com/ developers/voyager/white/voyager20.pdf.

PEINE, H., and STOLPMANN, T. (1997): The architecture of the Ara platform for mobile agents. *Proc. of the First International Workshop on Mobile Agents, Lecture Notes in Computer Science,* 1219:50-61, Berlin, Springer-Verlag.

RUS, D., GRAY, R., S., and KOTZ, D. (1997): Transportable information agents. *Proc. of International Conference on Autonomous Agents,* 228-236.

Swarm Design Group, A Tutorial Introduction to Swarm. Available at http://www.swarm.org/csss-tutorial/frames.html.

STRAßER M., BAUMANN, J., and HOHL, F. (1997): Mole - A Java based mobile agent system. *Special Issue in Object Oriented Programming*, M. Mühlhäuser (ed.), Dpunkt Verlag, 301-308.

THOMSEN, M., LETH, L., and PRASAD, S. (1993): Facile Antigua release programming guide. *Technical Report ECRC-93-20*, European Computer Industry Research Centre, Munich, Germany.

WHITE, J. E. (1997): Mobile agents. *Software Agents*, MIT Press.

ZHANG, M., and LI, W. (1999): Persisting autonomous workflow for mobile agents using a mobile thread programming model. *Approaches to Intelligent Agents, Lecture Notes in Artificial Intelligence*, 1733:84-95, Berlin, Springer-Verlag.

BIOGRAPHICAL NOTES:

Wei Li is Associate Professor in Computer Science, in the Department of Information Management, Capital University of Economics and Business, Beijing P.R. China. Currently he is a Post Doctoral Research Fellow in the Faculty of Informatics and Communication, Central Queensland University, Australia. He was awarded his Ph.D. degree in Computer Science in the Institute of Computing Technology (ICT), Chinese Academy of Sciences, in July of 1998. His major is in the area of Artificial Intelligence and Distributed Computation.

Minjie Zhang is a Senior Lecturer in the School of Information Technology and Computer Science, University of Wollongong, Australia. She received her Bachelor of Computer Science degree from Fudan University, China in 1982 and received her Ph.D. degree in Computer Science from the University of New England, Australia in 1996. She is a member of the IEEE, IEEE Computer Society, and ISCA. She is the author, or co-author, of over thirty research papers. Her research interests include distributed artificial intelligence, intelligent information retrieval, and software engineering.

A Meta-data Update Policy in a Flash Memory File System

Shihoon Cho and Joonwon Lee

Department of Computer Science, Korea Advanced Institute of Science and Technology,
373-1 Kusong-Dong, Yusong-Gu, Taejon 305-701, Korea
Email: shcho@camars.kaist.ac.kr, joon@camars.kaist.ac.kr

Joongsoo Ma

Information and Communication University,
58-4, Hwaam-Dong, Yusong-Gu, Taejon, 305-732, Korea
Email: jsma@icu.ac.kr

This paper describes a new mechanism for meta-data update in file systems using flash memory. To fulfill flash memory as a stable storage system, one of its main draw-backs, meta-data update, which occupies the majority of write operations, must be reduced. That is, the file system should avoid weaknesses of flash memory, such as a limited number of program/erase cycles and its inability of in-place updates. We exploit two types of i-nodes which are allocated in separate flash areas and also introduce a delta logging method to efficiently utilise flash memory space. A trace-driven simulation shows the results that the performance of our system with two workloads is improved in terms of both time and space utilisation.

Keywords: Operating Systems, Flash Memory, File System, Meta-data

1 INTRODUCTION

Traditional computer systems construct file systems using magnetic disks as secondary storage. However, magnetic disks are not recommended in the mobile computing environment because they consume lots of power and cannot guarantee reasonable resistance to shocks and vibration. Flash memory, a new type of solid state memory, has been replacing magnetic disks as a persistent storage system in the mobile environment since it provides useful characteristics, such as non-volatility, in-system programmability, low power consumption, reasonable tolerance, and high density (Douglis, Cáceres, Kaashoek, Li and Marsh, 1994; Kawaguchi, Nishioka and Motoda, 1995; Wu and Zwaenepoel, 1994).

On the other hand, flash memory also has drawbacks: *erase-before-rewritten, write access times higher than read access times,* and *a limited number of program/erase cycles.*

Accordingly, there are several obstacles to overcome when implementing a file system on flash memory rather than magnetic disks. Conventional file systems based on magnetic disks have an inherently **hot region** for meta-data manipulation (e.g., a FAT in MS-DOS or super-blocks/i-nodes in UNIX file system). If they are implemented on flash memory, their own in-place update schemes do not make sense at all. The erase-before-rewritten constraint of flash memory makes it difficult to update new information in its place. Previous studies indicate that *the majority of writes are for meta-data* and emphasise that *meta-data update is important to file system performance* (Ganger and Patt, 1994; Ruemmler and Wilkes, 1993; Vahalia, Gray and Ting, 1995).

Copyright© 2001, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the JRPIT copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Australian Computer Society Inc.

Manuscript received: September 2000, revised May 2001
Associate Editor: Hugh Williams

Meta-data updates, which frequently occur during file manipulation, impose overheads in spatial cost on the system since the overwritten region of the flash memory is forbidden. Furthermore, with UNIX-like file systems implemented on flash memory, the read operation of a file invalidates a whole i-node block for the file, and a new flash block for the i-node must be allocated because the file system updates the record of the last access time stamp for the file. Even bit-level meta-data updates require write operations on the entire data block.

In this paper, we propose a mobile file system with flash memory (MFWFM) as a substitute for Hard Disk Drives (HDDs). The primary approach of MFWFM is to efficiently reduce the problems associated with meta-data updates while avoiding flash memory's drawbacks as mentioned. Our MFWFM exploits a new organisation of a file system, offering the concept of in-place updates on flash memory and devises a delta logging method to reduce the amount of flash memory spaces incurred by frequent occurrences of meta-data updates. What is more, MFWFM separates an existing i-node into two types of i-nodes in order to increase the throughput of write operations and to improve system performance, presenting parallelism of write operations to separated areas.

The performance of proposed MFWFW is evaluated by a trace-driven simulation with realistic traces as well as synthetic traces.

The remainder of this paper is organised as follows: Section 2 describes the MFWFM and various techniques to support it. Section 3 evaluates MFMWM using trace-driven simulations, presents the results, and analyses them. Section 4 reviews several issues related to our MFWFM. Section 5 summarises our results and presents our conclusion.

2 MFWFM ARCHITECTURE

2.1 Overview

MFWFM architecture is logically composed of two layers: the upper layer and the lower layer. The upper layer, based on UNIX System V file system, supports system call interfaces and fetches information from lower layer facilities for file manipulations. The lower layer manages flash memory directly and is also divided into two sub-layers: DRAM layer and Flash layer. Figure 1 shows the layout of the lower layer in MFWFM.

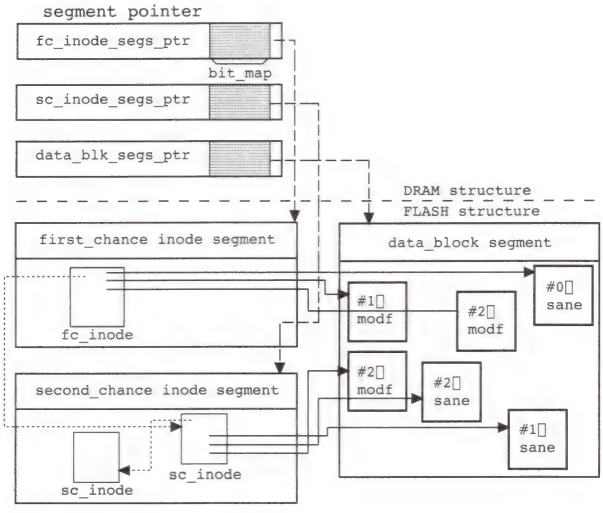


Figure 1: Layout of MFWFM lower layer

For indicating segments of a file, the DRAM layer maintains three types of in-core segment pointer structures: *fc_inode_segs_ptr, sc_inode_segs_ptr,* and *data_blk_segs_ptr*. When a file is initially created, the fc_inode_segs_ptr represents all of its segments.

After write operations for a file, modified segments are indicated by sc_inode_segs_ptr. The data_blk_segs_ptr stands for the list of data block segments.

Bitmaps within each pointer represents the utilisation of flash memory for free space management: *free_map* and *sane_map*. The former allows a fast searching of free blocks within the segment while the latter enables the segment manager to choose a victim segment in segment management processing.

The reason for holding the bitmaps in DRAM is due to the drawbacks of the flash memory. As we have stated before, flash memory cannot be updated in-place, and bit-wise operation in flash memory is unavailable.

Flash layer consists of three types of on-chip structures: first_chance_inode_segment, second_chance_inode_segment, and data_block_segment. All segments of a file can be arbitrarily mapped onto any of the three segments. Each segment contains its descriptor which represents information such as its segment type, number of erase cycles, and block usage.

2.2 Separation of i-nodes

MFWFM provides the concept of separation among i-nodes. Whenever data blocks of segments are modified or attributes of a file are changed, MFWFM logs the updates sequentially onto other i-node data structures. Contrary to HDDs, flash memory does not support in-place update. Hence, a new i-node is devised for processing the updated information.

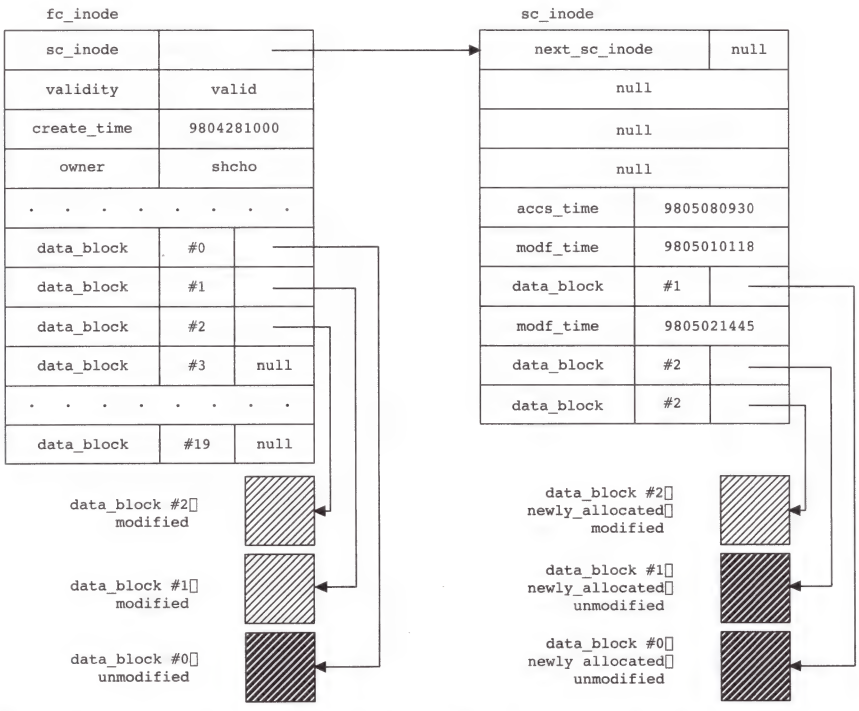


Figure 2: Relationship between first_chance_inode, second_chance_inode and data blocks

Once a file is created, a set of first_chance_inode, *fc_inode*, is defined. First_chance_node is identical to i-nodes in conventional UNIX, except it has a pointer to second_chance_inode, *sc_inode*. Second_chance_inode contains only the updated information, *delta logging*, which happens during file manipulations. Whenever applications deal with a file, MFWFM does not invalidate the entire block for the file, but only appends appropriate logs of the updated fields of fc_inode onto sc_inode. Data_block segment indicates data blocks related to each corresponding inodes. Figure 2 shows the relationships among first_chance_inode, second_chance_inode, and data blocks.

By means of separating the i-nodes, we can obtain parallelism of write operations. The write operations for both meta-data and data itself can be initiated sequentially but need not be processed synchronously. In other words, the work for data writes does not have to wait for the completion of their meta-data.

The latter is guaranteed by flash memory characteristics irrespective of power failure. Even with a power loss, the last modification information remains consistent because both meta-data and pure data are logged.

For the delta logging, each log is composed of *log-type* and *log-info*. Log-type is determined according to the updated attributes of a file, and the contents of log-info field contain the updated attributes. MFWFM provides various types of logs as follows:

- FC_INODE_LOG_TYPE : indicates the creation of a file.
- F_NAME_LOG_TYPE : indicates the changing name of a file.
- F_SIZE LOG_TYPE : indicates the changing size of a file.
- OWN_U_LOG_TYPE : indicates the changing user-ID of owner.
- OWN_G_LOG_TYPE : indicates the changing group-ID of owner.
- CRE_TIME_LOG_TYPE : stands for the i-node creation time.
- LAC_TIME_LOG_TYPE : stands for the time stamp of last access time.
- MDF_TIME_LOG_TYPE : stands for the changing of i-node modified time.
- DBLK_DIRECT_LOG_TYPE
 : indicates the changing of pointer for directly indexed data block.
- DBLK_SINGLE_LOG_TYPE and INDX_SINGLE_LOG_TYPE
 : indicates the changing of pointer for single indirectly indexed data block.
- DBLK_DOUBLE_LOG_TYPE and INDX_DOUBLE_LOG_TYPE
 : indicates the changing of pointer for double indirectly indexed data block.
- DBLK_TRIPLE_LOG_TYPE and INDX_TRIPLE_LOG_TYPE
 : indicates the changing of pointer for triple indirectly indexed data block.

2.3 Segment manager

Segment manager is invoked to reclaim the segments of flash memory when available segments for file operations are inadequate. For simplicity, MFWFM adopts a greedy policy, from which the segment manager selects the most invalidated segment for reclamation. Once the segment manager chooses a segment to be erased as a victim, all valid blocks of the segment are collected and copied onto a new segment. A command issued from MFWFM erases the victim segment and the erased segment is used for new data. Current flash hardware permits multiple segments to be erased simultaneously as long as each segment belongs to difierent banks.

2.4 Algorithms for File Operation

File Open and Close

The concept of open in MFWFM is to gather updated information from a set of logs in sc_inode and to construct complete i-node information. However, if the file being opened is a frequently accessed one, such as *hot file,* it may have too many logs to retrieve since a lot of sc_inodes may be attached to fc_inode for logging of i-node modification. Therefore, MFWFM limits the number of loggings for each file so that it gains reasonable latency when opening a file. The total time to open a file in MFWFM is shown below:

$$\begin{aligned} T_{open} &= T_{fetch_fc_inode} + T_{update_inode} \\ &= T_{fetch_fc_inode} + N_{log} * t_{log_type} \\ &+ n_{info} * t_{log_info} + n_{fc} * t_{fc_inode} \end{aligned}$$

In the above equation, $T_{fetch_fc_inode}$ is time to fetch fc_inode from flash memory and T_{update_inode} is the time to update fc_inode along with logs from sc_inode, where t_{log_type} is the time to read one log_type field from sc_inode, t_{log_info} is the time to read one $_{log_info}$ field from sc_inode, and t_{fc_inode} is the time to read one fc_inode field from sc_inode. N_{log} is the number of all logs and is calculated by $n_{info}+n_{fc}$, where n_{info} is the number of log_info type logs and n_{fc} is the number of fc_inode type logs.

Be that as it may, the total time to open a file in conventional file systems, such as FMFS, is diffierent from the equation shown above. T_{update_inode} is not used for calculation. As a result, the main impact of *open* is dependent upon the cost of T_{update_inode}. From segment manager's viewpoint, if the number of logs for each files exceeds the threshold that MFWFM offers, segment manager designates the particular segment as a victim for reclamation. While the segment manager cleans up the corresponding segment, all logs within the segment are collected and merged into a new fc_inode. Actually, even for files that contain a number of logs, the cost required for log gathering, T_{update_inode}, is not critical because the read access time in flash memory is comparable to those of DRAM.

Just as with UNIX, the last access time must be recorded when *closing* a file. However, time stamping in MFWFM has only to append a log which represents the change of the last access time. In contrast, conventional file systems invalidate the entire block for the i-node. The time to close a file is calculated as follows:

$$\begin{aligned} T_{close} &= T_{update_incore_inode} + T_{update_flash_inode} \\ &= T_{update_incore_inode} + t_{last_log} + t_{append_log} \end{aligned}$$

where $T_{update_incore_inode}$ is time to update incore inode within DRAM and $T_{update_flash_inode}$ is time to copy updated information for i-node into flash memory. t_{last_log} is time to search the last log by means of scanning sc_inodes and t_{append_log} is time to write a log into sc_inode.

In the above formula, t_{last_log} is a function of few nsec and t_{append_log} is dependent upon the type of log. If there is no room for logging in preallocated sc_inode, T_{close} requires additional cost which is needed for allocating a new sc_inode. Besides, if the number of logs exceeds the threshold, *close* procedure should invoke the segment manager instantly.

File Create and Delete

To *create* a file, MFWFM has to determine whether the corresponding fc_inode is being used or not. For initial creation, MFWFM has merely to set up appropriate fields on corresponding fc_inodes. Otherwise, MFWFM allocates a new sc_inode on the requested fc_inode, and establish fc_inode pointer to the sc_inode. When a given fc_inode is dirtied two or more times, MFWFM may probably contain one or more sc_inodes. At best, the overhead for file creation is identical to that of FMFS. However, MFWFM requires additional cost for allocating a new sc_inode and appending the log in the worst case.

Delete algorithm in MFWFM is simple. First of all, MFWFM searches for the last fc_inode and marks its fc_validity field as INVALID. All indices for data blocks remain as they are when deleting the i-nodes but sane_map is marked as zero. Reclamation procedure of segment_manager uses its bit vector.

File Read/Write and Other Operations

Read algorithm in MFWFM is identical to read operations in traditional UNIX implementation. Since in-core i-node is already updated during open operation, read operations use complete information from in-core i-node.

The algorithm for *write* is more complicated than the traditional UNIX implementation. We consider three types of write operations. In the case of modifying an allocated data block, MFWFM allocates a buffer cache in DRAM space for update in advance. MFWFM copies original data from the flash data block to buffer cache and modifies the corresponding data flock within the buffer cache. After the modification of data block, the contents of the buffer cache are written back to the flash memory area which is newly allocated. The index for the new data block must be recorded in sc_inode. In the case of appending new data to a file, write operations allocate new data blocks. MFWFM records their address into appropriate fields in fc_inode. The other case for write operation is to modify or *truncate* the size of a file. To truncate a file by a given size, MFWFM invalidates only the appropriate log instead of the entire i-node block and then appends such log to last log in sc_inode.

To position a file pointer to a specified offset, MFWFM need only establish the current file pointer in in-core i-node. As stated before, in-core i-node acquires complete information during the open operation. As for *seek* operation, the current file pointer need not be recorded to actual i-node in flash memory. To change certain attributes of an i-node, MFWFM have to only append an appropriate log to a set of logs.

Factor	Specification
Flash Memory Size	64 MBytes
Number of Segment	16
Segment Size	128 KBytes
Segment Erase Time	1.2 sec
Byte Program Time	6 μsec
Page Size	256 Bytes
Page Program Time	1 msec
Read Access Time	30 nsec
Maximum Read Throughput	33.33 MBytes/s
Maximum Write Throughput	244.1 KBytes/s
Erase Throughput	106.7 KBytes/s

Table 1: Flash Memory Component specification based on Intel 28F016XS

3 PERFORMANCE EVALUATION AND DISCUSSION

For simulation, we assume a bus-based simulation architectural model that is composed of a CPU, a set of DRAMs, a set of flash memory, and two auxiliary I/O devices.

3.1 Simulation Methodology

To study the performance implications, we developed two trace-driven simulators which run on CSIM (Schwetman, 1992).

One of the two implemented simulators is for simulating our designed MFWFM and the other is for simulating comparative file system model FMFS (Kawaguchi, *et al*, 1995).

Table 1 represents the parameters of flash memory used in the simulation. Of all 64 MBytes of MFWFM space, the default portion for fc_inode type segment, sc_inode type segment, and data_block type segment is defined by 7%, 3% and 90% respectively. Unfortunately, since sufficient space, as with HDDs, cannot be reserved for file system spaces, our simulator cannot allocate actual file space for real traces.

Consequently, we need new synthetic traces and also manage the file descriptor information to the full extent in order to evaluate the throughput of read/write operations and to measure the amount of invalidated flash memory during realistic file operations.

The simulators are driven by two kinds of workloads: *real traces* and *synthetic traces*. The former is the Sprite file system traces which were collected from a real environment (Hartman, 1993).

Using the Sprite file system traces, we can evaluate the performance of MFWFM under a realistic file system environment. In the case of synthetic traces, file size information was generated based on the result of UNIX *file size survey* (Irlam, 1994).

We assume the average interarrival time of each event in synthetic traces as 500 μsec, while the average of inter-arrival time in the Sprite file system traces is about 17.5 msec.

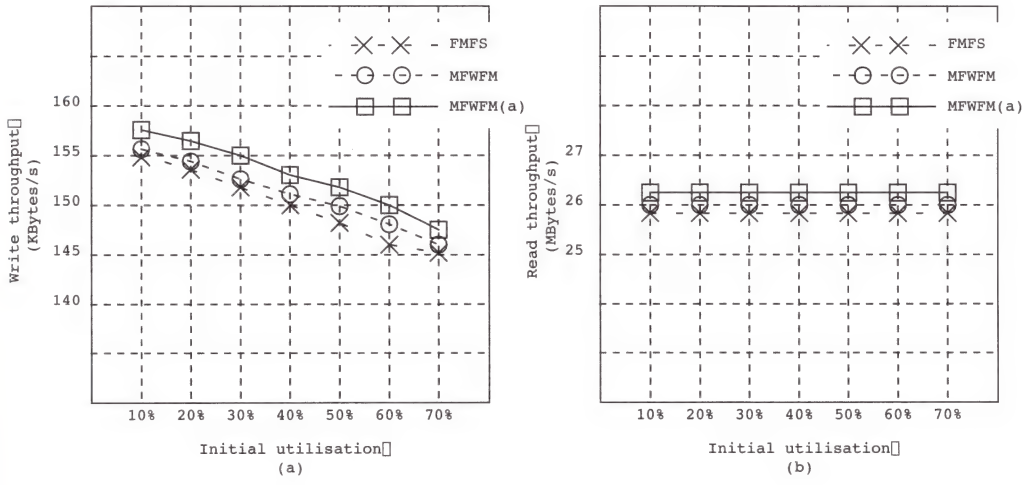


Figure 3: Effect of read/write performance based in initial utilisation

3.2 Results and Analysis

3.2.1 Simulation Using Synthetic Traces

The purpose of the first experiment is to test the sequential write performance to measure maximum throughput. In these traces, as for file systems filled with fixed initial data, a certain amount of data was repeatedly written. The comparison of sequential write performance in FMFS, synchronised MFWFM and asynchronised MFWFM is shown in Figure 3(a).

The curves show the result based on difierent initial data. Simulation results for FMFS show that as the amount of initial data gradually increases, the throughput decreases. Despite this there is no need to invalidate data blocks for writing, the time to find free data block will gradually increase according to the utilisation of the file system. The difierence between two types of MFWFM is to tell whether operation to write a data waits for the completion of write to an appropriate meta-data or not. The tiny improvement in the *asynchronised*–MFWFM is derived from reducing the time for recording a data block index into i-node.

In MFWFM, read operations have only to fetch the address of data block with indices. Since log for indices is already updated during a file open, there is no need to find free block and to invalidate the flash memory. Unlike a HDD-based file system, since MFWFM is free from the overhead of *seek latency* and *rotational delay*, the results of read operations in traces have uniform values, regardless of the policy for block layout or file system utilisation. Figure 3(b) shows that the average read throughput is nearly 26 MBytes/s and provides the overall improvement of read performance against FMFS.

This improvement is due to the reduction of remapping time. That is, in the case of MFWFM, the address of data block in i-node is the real address of flash data block. On the other hand, the block number in i-node should be remapped into references of the translation table in FMFS.

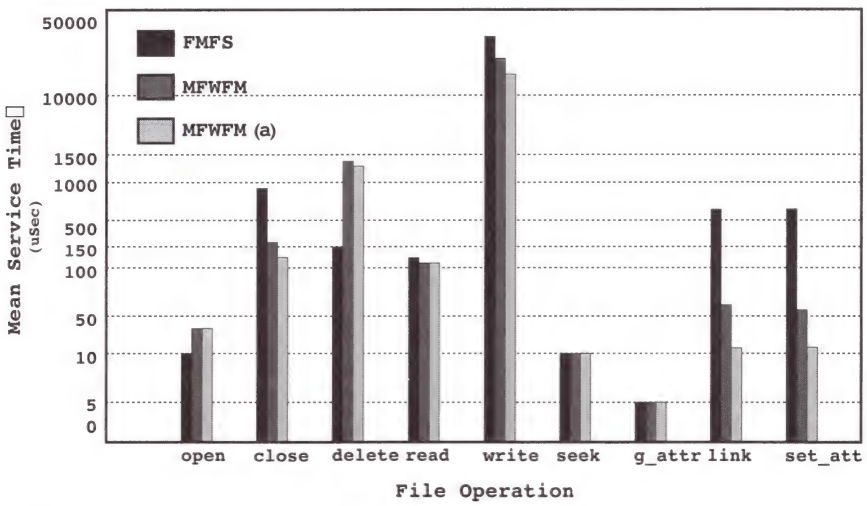


Figure 4: File operation performance under real traces

3.2.2 Simulation using Realistic Workloads

The purpose of simulation using real traces is to observe performance improvement in file operations and to measure the effects of flash memory invalidation. Figure 4 summarises the mean service time for each file operation. From the simulation results, we can observe that the cost of open operation in MFWFM is four times greater than that of open in FMFS and MFWFM provides large variance for open operation. Open in FMFS should reference the translation table to remap the address of i-node block into flash memory and it actually requires a copy to read i-node block from flash memory. However, MFWFM requires additional cost to update i-node with logs for open operation.

In contrast, the cost of *close* operation in FMFS is almost seven times higher than that of close operation in MFWFM, since close operation in FMFS requires write time for the entire i-node block even for modification of just one field, such as *last access time*. On the other hand, MFWFM need only write the appropriate log, LAC_TIME_LOG_TYPE. The costs for *read/write* operations are similar to the results of simulation using synthetic traces. In FMFS, the costs for *mk_link* and *set_attr* operations have identical values because they require full remapping of i-node block for the modification of each field. The cost of *get_attr* and *seek* operation represents the same value in all cases.

Table 2 shows the comparison of throughput in read/write operations and the amount of invalidated flash memory. The result shows that the amount of invalidated flash memory in MFWFM is reduced by 14.75 % against FMFS.

Model	*read throughput*	*write throughput*	*invalidated Bytes*	*pure data ratio*
FMFS	28.245133	0.155778	998670336	0.939296
MFWFM	28.924958	0.158302	870261296	0.972458
MFWFM(a)	28.924958	0.158668	870261296	0.972458

Table 2: Comparison of throughput in read/write operations and amount of invalidated flash memory under real traces : unit in throughput is average MBytes/s

4 Related Work

There has been numerous work done inanumber of fields related to our interests. Meta-data logging (Vahalia, *et al*, 1995) for NFS server and meta-data update (Ganger and Patt, 1994) supporting ordered writes aimed to handle crash recovery. They tried to deal with meta-data operation efficiently, maintaining the integrity of the file system on-disk structures. Furthermore, IBM's System R (Astrahan *et al*, 1976, Gray *et al*, 1981) used the transaction log and the shadow-file mechanism to facilitate application specific recovery for medium failure. Srinivasan *et al* (1985) designed the shadow-file concept coupled with the check point for recovery in microprocessor systems.

Douglis *et al* (1994) examined three storage alternatives for mobile computing. They found that at a 90 % utilisation or above, of large erasure units might result in performance degradation.

Cáceres *et al* (1993) emphasised the operating system's role to increase read throughput as well as to reduce write performance with battery-backed DRAM.

eNVy (Wu and Zwaenepoel,1994) presented its storage space as a linear and memory mapped array and found it was actually a large non-volatile main memory storage system rather than a file system.

To overcome flash memory's inability of in-place update and high write cost, segment clean or data reclamation was provided in log-structured file systems (LFS) (Robinson, 1996; Rosenblum and Ousterhout, 1992; Seltzer, Bostic, Mckusick and Staelin, 1993).

Kim *et al* (1999) proposed a new flash memory management based on a log structured file system, focusing on lowering the cleaning cost and evenly utilising flash memory cells. With the collection operation of cold and non-cold data, they expected to reduce the total cleaning cost compared to greedy policy.

MS-Flash (Intel Corporation, 1995) was provided as a MS-DOS compatible file system using flash memory cards. For flash areas that were overwritten frequently, MS-Flash handled them by a linked-list data structure. However, even for byte-level modifications, since the whole space for the file should be invalidated, the overhead of meta-data update has significantly exerted a negative influence on the file system.

Kawaguchi *et al* (1995) proposed a flash memory based file system (FMFS) that supports UNIX file system transparency and segment cleaning. It was executed by way of LFS to minimise flash-memory's write cost. Whenever data blocks were overwritten through write operations, FMFS invalidated all of them and issued several commands to reclaim the invalidated data blocks. Therefore, whether write operations were originated from user data blocks or meta-data, FMFS intrinsically imposed overheads in spatial cost as utilisation of flash memory increased.

From a hardware point of view, a lot of studies have been done to make the best of flash memory in applications such as solid state disks and mobile computing. Khouri *et al* (2000) proposed a fast read word-line and program bit-line voltage regulator for multilevel flash memories. Low power structure added to the basic circuit was devised for high recovery speed.

Mohammad *et al* (2000) provided algorithms for handling faults specific to flash memories. They exploited test algorithms to detect three types of disturbances in flash memories.

5 Conclusion

Flash memory is becoming an increasingly important medium of persistent storage in the mobile environment. Characteristics, such as high density, non-volatility, and low power consumption, flash memory makes it the ideal storage system, replacing the conventional HDD's place in mobile computing. However, flash memory should overcome its inability of in-place updates in order to be used as a stable storage system.

In this paper, we proposed a new file system, MFWFM, that provides traditional UNIX system call transparency based on flash memory. To minimise the amount of meta-data update as much as possible, MFWFM separated conventional i-node into two types of i-nodes: fc_inode and sc_inode. Each time when an application updates contents or fields of a file, MFWFM does not invalidate the entire block but appends only appropriate logs onto sc_inode. To maintain a sufficient number of free blocks, segment manager collects insane segments. Even while update of meta-data is progressing, MFWFM allows other applications to update their own data. Our simulation results provide evidence that MFWFM outperforms conventional FMFS.

However, a number of interesting works still remain, such asthe modelling of the likely uses of mobile computing environment and the benchmarking for various realistic sized flash memory.

REFERENCES

ASTRAHAN, M.M; BLASGEN, M.W., CHAMBERLIN, D.D., ESWARAN, K.P., GRAY, J.N., GRIFFITHS, P.P., KING, W.F., LORIE, R.A., McJONES, P.R., MEHL, J.W. PUTZOLU, G.R, TRAIGER, I.L., WADE, B.W. and WATSON, V.

(1976): System R: Relational approach to database management, *ACM Transactions on Database Systems*, 1 (2): 97-137.

BAKER, M., ASAMI, S., DEPRIT, E., OUSTERHOUT, J. and SELTZER, M. (1992): Non-volatile memory for fast, reliable file systems, *Proceedings of the Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, Boston, MA, 10-22.

CÁCERES, R., DOUGLIS, F., LI, K. and MARSH, B. (1993): Operating system implications of solid-state mobile computers, *Proceedings of the Fourth Workshop on Workstation Operating Systems*, Napa, California, 21-27.

DOUGLIS, F., CÁCERES, R., KAASHOEK, F., LI, K., MARSH, B. and TAUBER, J.A. (1994): Storage alternatives for mobile computers, *Proceedings of the first USENIX Symposium on Operating Systems Design and Implementation*, Monterey, California, 25-37.

FORMAN, G.H. and ZOHORJAN, J. (1994): The challenges of mobile computing, *UW CSE Tech Report*, 93:11-03.

GANGER, G.R. and PATT, Y.N. (1994): Metadata update performance in file systems, *Proceedings of the first USENIX Symposium on Operating Systems Design and Implementation*, Monterey, California, 49-60.

GRAY, J., McJONES, P., BLASGEN, M., LINDSAY, B., LORIE, R., PRICE, T., PUTZOLU, F. and TRAIGER, I. (1981): The recovery manager of the system R database manager, *Computing Surveys*, 13 (2): 223-242.

HARTMAN, J.H. (1993): Using the Sprite file system traces, *http://now.CS.Berkeley.EDU/Xfs/SpriteTraces/docs/sospTraces.ps.gz*.

INTEL CORPORATION. (1995): *Flash memory volume 1-2*.

IRLAM, G. (1994): UNIX file size survey-1993, *http://www.base.com/gordoni/ufs93.html*.

KAWAGUCHI, A., NISHIOKA, S. and MOTODA, H. (1995): A Flash-memory based file system, *Proceedings of the Winter USENIX 1995 Technical Conference*, New Orleans, Louisiana, 155-164.

KHOURI, O., MICHELONI, R., GREGORI, S. and TORELLI, G. (2000): Fast voltage regulator for multilevel Flash memories, *Proceedings of the 2000 IEEE International Workshop on Memory Technology, Design and Testing*, 34-38.

KJELSO, M. and JONES, S. (1995): Memory management in Flash-memory disks with data compression, *LNCS Volume 986*, Springer Verlag, ISBN 3-540-60368-9, 399-413.

KIM, H.J. and LEE, S.G. (1999): A new Flash memory management for Flash storage system, *Proceedings of the 23rd International*. Computer Software and Applications Conference, 294- 299.

LEE, Y.W., LEUNG, K.S. and SATYANARAYANAN, M. (1999): Operation-based update propagation in a mobile file system, *1999 USENIX Annual Technical Conference*, Monterey California, 43-56.

MCKUSICK, M., JOY, W., LEFFLER, S. and FABRY, R. (1984): A fast file system for UNIX, *ACM Transactions on Computer Systems*, 2(3): 181-197.

MOHAMMAD, M.G., SALUJA, K.K. and YAP, A. (2000): Testing Flash memories, *Proceedings of the 13th International Conference on VLSI Design*, Calcutta, India, 406-411.

ROBINSON, J.T. (1996): Analysis of steady-state segment storage utilisation in a log-structured file system with least-utilised segment cleaning, *Operating System Review*, 30(4): 29-32.

ROSENBLUM, M. and OUSTERHOUT, J.K. (1992): The design and implementation of a log-structured file system, *ACM Transactions on Computer Systems*, 10(1): 26-52.

RUEMMLER, C. and WILKES, J. (1993): UNIX disk access patterns, *Proceedings of the Winter USENIX 1993 Technical Conference*, San Diego, California, 405-420.

SCHWETMAN, H. (1992): CSIM Reference Manual (Revision 16).

SCHWETMAN, H. (1992): CSIM Users' Guide (Revision 16).

SEE, D. and THURLO, C. (1994): Managing data in an embedded system utilising Flash memory, *Intel Technical Note*.

SELTZER, M., BOSTIC, K., MCKUSICK, M.K. and STAELIN, C. (1993): An implementation of a log-structured file system for UNIX, *Proceedings of the Winter USENIX 1993 Technical Conference*, San Diego, California, 307-326.

SRINIVASAN, B. and GUNASINGHAM, H. (1985): Recoverable file system for microprocessor systems, *Microprocessors and Microsystems*, 9 (4): 179-183.

VAHALIA,U., GRAY, C.G. and TING, D. (1995): Metadata logging in an NFS server, *Proceedings of the Winter USENIX 1995 Technical Conference*, New Orleans, Louisiana, 265-276.

WU, M. and ZWAENEPOEL, W. (1994): eNVy: A non-volatile, main memory storage system, *Proceedings of the Sixth International Conference on Architectural Support for Programming Languages and Operating Systems*, San Jose, CA, 86-97.

BIOGRAPHICAL NOTES

Shihoon Cho received his B.S. degree in computer science at Kyungpook National University and his M.S. degree in information and communication engineering at KAIST. From 1996 to 2001 he was with the Wireless Communication Division of Samsung Electronics Co. Ltd. He is currently a Ph.D. candidate in the division of computer science of the department of electrical engineering and computer science at KAIST. His research interests include file systems, operating systems, computer architecture and mobile communication.

Joonwon Lee earned his B.S. degree in statistics and computer science at Seoul national University and M.S. and Ph.D. degrees in computer science at the College of Computing, Georgia Institute of Technology. At IBM, he worked in a multiprocessor design team. He is currently an associate professor in the division of computer science of the department of electrical engineering and computer science at KAIST. His research interests are operating systems, computer architecture and parallel processing.

Joongsoo Ma received his M.S. and Ph.D. degrees in electrical and computer engineering from the University of Massachusetts, Amherst in 1977 and 1978, respectively. From 1978 to 1991 he was a research staff member at the IBM T.J. Watson research centre, working on protocol analysis and design. From 1991 to 1998 he was a managing director at the SK Telecom research centre, Seoul, Korea, working on mobile network design and management. Since 1999 he has been a professor at Information and Communications University, Taejon, Korea. His current research areas include design and analysis of wireless communications protocols.

The Effects of Normalisation on the Ability of Business End-Users to Detect Data Anomalies: An Experimental Evaluation

A. F. Borthick
Georgia State University
School of Accountancy
Atlanta, Georgia 30302-4050 USA
Internet: borthick@gsu.edu

P. L. Bowen
Department of Commerce
The University of Queensland
Brisbane, Queensland, Australia 4072
Internet: bowen@lorien.commerce.uq.edu.au

M. R. Liu
Optus
Level 8, 50 Miller Street
North Sydney, NSW, 2060

F. H. Rohde
Department of Commerce
The University of Queensland,
Brisbane, Queensland, Australia 4072
Internet: rohde@commerce.uq.edu.au

With the proliferation of relational database programs for PCs and other platforms, many business end-users are creating, maintaining, and querying their own databases. More importantly, business end-users use the output of these queries as the basis for operational, tactical, and strategic decisions. Inaccurate data reduce the expected quality of these decisions. Implementing various input validation controls, including higher levels of normalisation, can reduce the number of data anomalies entering the databases. Even in well-maintained databases, however, data anomalies will still accumulate. To improve the quality of data, databases can be queried periodically to locate and correct anomalies. This paper reports the results of two experiments that investigated the effects of different data structures on business end-users' abilities to detect data anomalies in a relational database. The results demonstrate that both unnormalised and higher levels of normalisation lower the effectiveness and efficiency of queries relative to the first normal form. First normal form databases appear to provide the most effective and efficient data structure for business end-users formulating queries to detect data anomalies.
(Keywords: End-user queries, Data quality, Normalisation)

1. INTRODUCTION

Since the late 1980s, the availability and usability of relational database programs has significantly increased, especially for mini- and personal computers. This increase has led to a proliferation of databases being created and maintained by business end-users. These databases are often queried to provide information for important business decisions. The quality of these decisions depends on the

Copyright© 2001, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the JRPIT copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Australian Computer Society Inc.

Manuscript received: May 2000, revised August 2001
Associate Editor: Graham Low

quality of the data, i.e., the value and usefulness of information systems depend on the accuracy of their data (Janson, 1988; Redman, 1995; Kaomea and Page, 1997). Empirical studies of corporate databases have shown that many organisations' databases contain anomalies (Morey, 1982; Laudon, 1986; Icerman and Hillison, 1991; Medawar, 1995). Information technology (IT) professionals designed these corporate databases and their controls. Databases designed by business end-users, who have limited or no training in relational databases, are likely to contain an even larger percentage of anomalies.

Relational database theory has demonstrated that higher levels of normalisation reduce data redundancy (Codd, 1970; Brookshire, 1993; Elmasri and Navathe, 2001; Date, 2000; Date, 1998). This reduction in redundancy minimises data anomalies introduced while inserting, updating, or deleting data elements or records. Normalisation eliminates repeating groups and resolves lack of atomicity. Atomic fields contain only one piece of data, e.g., a single discount field that contains both the discount percent and the number of days to which the discount applies violates atomicity. The presence of repeating groups and non-atomic data attributes makes queries more complex. Unfortunately, decreased data redundancy resulting from higher levels of normalisation leads to an increase in data fragmentation, i.e., further normalisation generates larger numbers of tables with a smaller number of attributes in each table. This increased fragmentation leads to increases in query complexity as more joins and associated subqueries are required. Thus, increasing the level of normalisation may adversely affect business end-users' ability to formulate queries.

Because of data quality concerns, business end-users may need to query databases to detect anomalies. Little empirical research has examined the effects of various levels of normalisation on business end-users' ability to develop such queries. Through two experiments, this study examines the ability of business end-users to query a typical business database to locate data anomalies. The first experiment examines business end-user queries for three sets of data structures: not normalised (¬NF), first normal form (1NF), and third normal form (3NF). These three data structures match the two structures often observed in professionally developed databases (3NF), business end-user developed databases (¬NF), and a viable alternative (1NF). The second experiment focuses on differences in queries with 1NF and 3NF data structures.

Typically, business end-users design and create databases by focusing on access to data without regard to relational database theory. The resulting data structure may contain multi-valued data fields and redundant data, i.e., ¬NF. Business end-users often do not understand how multi-valued data fields and redundant data lead to insert, update, and delete anomalies, i.e., how their database designs lead to data quality problems. If they experience difficulty designing a database, business end-users can seek the help of IT professionals, but these professionals are likely to design databases to minimise data redundancy, e.g., by normalising to 3NF, but at the expense of ease of querying. Thus, unaided business end-users produce unnormalised data structures while database professionals produce third normal form or higher data structures. Intermediate data structures such as 1NF may, however, be more appropriate for business end-user queries.

2. EFFECTS OF NORMALISATION ON LOCATING DATA ANOMALIES

2.1 Data Accuracy and Data Quality

Ideally, organisations would maintain data quality by preventing anomalies from entering databases (Huh *et al.*, 1990). Organisations can reduce anomalies by creating well-designed databases and implementing extensive input validation controls. Normalisation of data structures, typically to third normal form or beyond, helps minimise the occurrence of insert, update, and delete anomalies. Good input validation controls also help minimise the possibility of data inaccuracies occurring

within a database. Unfortunately, even well-designed databases with extensive input validation controls do not result in completely accurate data. To avoid making decisions based on erroneous data, anomalies in databases need to be detected and corrected. One way to locate and correct data errors in a database is to query it periodically. As we will show, the underlying structure of the data, i.e., the normalisation level, affects the form, content, and complexity of these queries.

2.2 Normalisation

Normalisation is an elimination procedure (Codd, 1970) that gives successively more desirable levels of structural characteristics (Date, 2000). For a relation to be in first normal form (1NF), no attribute in the relation can contain more than one data value (Date, 2000). When a data structure moves from unnormalised (¬NF) to first normal form (1NF), all repeating fields are eliminated and non-atomic data items are decomposed into multiple atomic elements, i.e., all multi-valued attributes are resolved. When the level of normalisation increases from 1NF to 3NF, all partial and transitive dependencies are eliminated. The removal of repeating fields, non-atomic data items, and partial and transitive dependencies increases the fragmentation of the database, i.e., increases the number of tables. Other normal forms include Boyce-Codd normal form, fourth normal form, and fifth normal form. Because the experiments used only ¬NF, 1NF, and 3NF structures, Boyce-Codd (BCNF), fourth (4NF), and fifth (5NF) normal forms are not discussed here.

2.3 Relationship Between Normalisation and Query Complexity

Repeating fields and non-atomic data make queries of unnormalised data structures more complex. To retrieve data from data structures containing repeating fields requires numerous AND and OR operators in the condition clause of the query, i.e., in the WHERE clause of an SQL query. Non-atomic attributes increase query complexity because of the additional string manipulation functions required to extract specific data items. Data structures normalised to 1NF reduce the complexity of queries by eliminating repeating fields and non-atomic attributes. Although a slight increase in fragmentation occurs, compared with ¬NF, queries of 1NF data structures contain fewer terms and operators. The following information request with queries for ¬NF and 1NF data structures illustrates how multi-valued attributes make queries more complex.

List the items and corresponding receipt numbers where the payment date is earlier than the receipt date plus half the number of term days.

¬NF	1NF
SELECT *itemno, recrepno1, recrepno2* FROM ITEMA WHERE *paydate1* < (*recdate1*+(0.5*(TO_NUMBER (SUBSTR(*terms*,3,INSTR(*terms*,'/')+2))))) OR *paydate2* < (*recdate2*+(0.5*(TO_NUMBER (SUBSTR(*terms*,3,INSTR(*terms*,'/')+2)))));	SELECT ITEMB.*itemno, recrepno* FROM ITEMB, RECEIPTSB WHERE ITEMB.*itemno*=RECEIPTSB.*itemno* AND *paydate* < (*recdate*+(0.5**termdays*));

Databases normalised to higher normal forms, e.g., 3NF, often contain substantially more tables than lower normal forms, e.g., 1NF. To perform useful queries on these data structures, the data have to be accessed through join operations (Beynon-Davies, 1992). Therefore, although they eliminate update anomalies, higher normal forms often require additional joins to retrieve the desired data

(Date, 1986). The following information request with queries for 1NF and 3NF data structures illustrates this increase in complexity.

List the items and corresponding receipt numbers where the payment date is earlier than the receipt date plus half the number of term days.

1NF	3NF
SELECT ITEMB.*itemno, recrepno* FROM ITEMB, RECEIPTSB WHERE ITEMB.*itemno*=RECEIPTSB.*itemno* AND *paydate* < (*recdate*+(0.5**termdays*));	SELECT ITEMC.*itemno*, RECEIPTSC.*recrepno* FROM ITEMC, VENDORC, RECEIPTSC, PAYMENTSC WHERE ITEMC.*vno*=VENDORC.*vno* AND RECEIPTSC.*itemno*=ITEMC.*itemno* AND PAYMENTSC.*recrepno*=RECEIPTSC.*recrepno* AND *paydate* < (*recdate*+(0.5**termdays*));

As the complexity of the task increases, i.e., formulating queries in ¬NF or 3NF rather than 1NF data structures, the resulting cognitive demands on individuals increase (Campbell, 1988).

3. HYPOTHESES

3.1 Effectiveness

Databases in ¬NF contain non-atomic data items, repeating fields, or both. Data structures in 1NF are free of repeating fields and non-atomic attributes. Relative to 1NF, the structural problems associated with ¬NF data structures increase query complexity. Higher query complexity decreases business end-users' ability to formulate effective queries, e.g., users of ¬NF data structures are less able to formulate effective queries to locate data anomalies than users of 1NF data structures. This leads to the following hypothesis:

Hypothesis 1a: 1NF users will find more data anomalies than ¬NF users.

Relative to 1NF data structures, data in 3NF are fragmented into more tables and require users to join more tables to retrieve the desired information. Hence, relative to 1NF, queries in 3NF will be more complex. As before, higher complexity decreases a business end-user's ability to formulate effective queries. This leads to the following hypothesis:

Hypothesis 1b: 1NF users will find more data anomalies than 3NF users.

Compared to 3NF, queries of ¬NF databases require fewer joins as all data items are located in fewer tables, perhaps only one. ¬NF users, however, must contend with repeating fields, non-atomic data, and redundant data. These structural problems with ¬NF databases increase query complexity more than the fragmentation associated with 3NF databases. The following hypothesis represents business end-users' decreasing ability to formulate effective queries as query complexity increases:

Hypothesis 1c: 3NF users will find more data anomalies than ¬NF users.

3.2 Efficiency

A common approach to coping with complexity is to use a divide-and-conquer strategy. Users may attempt to circumvent the difficulties associated with ¬NF by developing multiple queries rather than a single query to retrieve the desired data. This strategy will generate more queries to detect

the same number of data anomalies. Furthermore, due to query anomalies associated with higher complexity, a higher percentage of ¬NF queries will fail to produce the desired results. Thus, more ¬NF queries will be required to obtain the same results as 1NF queries. This leads to the following hypothesis:

Hypothesis 2a: 1NF users will find more anomalies per query than ¬NF users.

Similarly, relative to 1NF, 3NF users may circumvent the additional complexity associated with data fragmentation by using multiple queries. This strategy will generate more queries to obtain the desired information. Alternatively, due to a greater probability of making query mistakes because of the higher complexity, a higher percentage of 3NF queries will fail to produce the desired results. This failure will require users to make more attempts to construct a query that produces the desired results. Thus, more 3NF queries will be required to obtain the same results as 1NF queries. This leads to the following hypothesis:

Hypothesis 2b: 1NF users will find more anomalies per query than 3NF users.

The non-atomic data items and repeating fields associated with ¬NF increase complexity more than the data fragmentation associated with 3NF. Hence, ¬NF users will experience more difficulty formulating queries than 3NF users. Thus, more ¬NF queries will be required to obtain the same data than 3NF queries. This leads to the following hypothesis:

Hypothesis 2c: 3NF users will find more anomalies per query than ¬NF users.

3.3 Questions Attempted

More complex queries require more time to formulate, i.e., users will need to spend more time retrieving data to satisfy individual information requirements. Because querying a ¬NF data structure is more complex than querying a 1NF data structure, users of ¬NF data structures will need to spend more time retrieving data to satisfy the individual information requirements. Thus, overall, users of ¬NF data structures are likely to investigate fewer questions than users of 1NF data structures. This leads to the following hypothesis:

Hypothesis 3a: 1NF users will investigate more questions than ¬NF users.

Because querying a 3NF data structure is more complex than querying a 1NF data structure, users of 3NF data structures will need to spend more time retrieving data to satisfy the individual information requirements. Thus, overall, users of 3NF data structures are likely to investigate fewer questions than users of 1NF data structures. This leads to the following hypothesis:

Hypothesis 3b: 1NF users will investigate more questions than 3NF users.

Because querying a ¬NF data structure is more complex than querying a 3NF data structure, users of ¬NF data structures will need to spend more time retrieving data to satisfy the individual information requirements. Thus, overall, the users of ¬NF data structures are likely to investigate fewer questions than users of 3NF data structures. This leads to the following hypothesis:

Hypothesis 3c: 3NF users will investigate more questions than ¬NF users.

4. RESEARCH METHOD: EXPERIMENT 1

4.1 Research design, participants, and data collection

This study used a laboratory experiment to test the hypotheses. Participants composed and executed queries in SQL for an Oracle database in one of three normalisation levels shown in Appendix A.

With few exceptions, e.g., Chan *et al.*, (1999), traditional approaches to testing query performance have used pencil and paper or simulated systems that do not provide the interactive feedback of typical organisational databases (e.g., Davis, 1990; Suh and Jenkins, 1992; Chan *et al.*, 1993). The user interface for this experiment was more realistic because participants received interactive feedback, both syntactic error messages and syntactically correct query results, from the system. Furthermore, they could correct and re-enter their queries as many times as they wished.

Eighty advanced undergraduate and master's level commerce and information technology students participated in the experiment. All participants were familiar with general computing concepts and activities and, prior to the experiment, had received training in developing SQL queries and data integrity controls. The experiment required participants to locate data inconsistencies or potential data inconsistencies in a just-in-time (JIT) II inventory system (Borthick *et al.*, 1998). Each participant examined one of three data structures: not normalised (¬NF), first normal form (1NF), or third normal form (3NF). All participants received a set of instructions containing the scenario and the details of tasks to be performed (Appendix B). Participants had two hours to search for inconsistencies in their assigned database by constructing appropriate queries for as many of the 12 questions as they could. Queries for these questions appear in Appendix C. Participants created script and log files to record their sessions, which were the source for the data for testing the hypotheses.

To eliminate any experience effect, participants were assigned randomly to one of the three treatment groups according to their information systems background. Based on the information provided by each participant, a knowledgeable IS person ranked each participant in ascending order, i.e., the person with the most extensive IS background was ranked 80 and the next ranked 79, etc. The order of the three treatment groups was then randomised, with the order being: ¬NF, 3NF, and 1NF. Participants were assigned according to their rank, i.e., 80 to ¬NF, 79 to 3NF, 78 to 1NF, 77 to 1NF, 76 to 3NF, 75 to ¬NF, etc. This method of randomisation was intended to make the IS experience and training of the groups as equivalent as possible.

4.2 Measures

The independent variable for all hypotheses was the level of normalisation: ¬NF, 1NF, or 3NF. The three dependent variables were the number of anomalies found, the number of anomalies found per query, and the number of questions attempted. To determine the number of anomalies found, the anomalies found by participants were matched with the list of anomalies seeded into the databases. The number of anomalies found per query was calculated by dividing the total number of anomalies found by a participant by the total number of queries executed by that participant. The number of questions attempted was determined by examining the log files.

5. RESULTS: EXPERIMENT 1

5.1 Summary Results

Table 1 shows a summary of the results from experiment 1. For each of the measures, the 1NF group performed best followed by the 3NF and ¬NF groups, respectively.

Model	Mean	Std Dev	Count (N)
Number of Anomalies Found			
¬NF	5.3462	2.9113	26
1NF	8.2917	3.5322	24
3NF	7.2308	3.2779	26
Number of Anomalies Found/Query			
¬NF	0.4838	0.2595	24
1NF	1.0056	0.3210	21
3NF	0.8594	0.4160	23
Percent of Actual Anomalies Found			
¬NF	0.4508	0.2497	24
1NF	0.5360	0.2240	21
3NF	0.4804	0.1925	23
Number of Questions Attempted			
¬NF	4.7083	1.5737	24
1NF	6.0476	1.6127	21
3NF	5.6522	1.2288	23

Table 1: Summary of results

5.2 Effects of Normalisation on Query Effectiveness

Hypothesis 1 predicted a direct relationship between the level of normalisation and the number of data anomalies found. The results of an analysis of variance (ANOVA) provided significant support for this hypothesis ($R^2 = 0.1274$, $F_{2,73} = 5.33$, $p = 0.0069$, two-tailed test). The least-squares means results (Table 2) show the presence, magnitude, and direction of differences between the three normal forms. The least-squares means results support the relationship predicted by Hypothesis 1a, i.e., 1NF users found significantly more anomalies than ¬NF users. The results also support Hypothesis 1c, i.e., 3NF users found significantly more anomalies than ¬NF users. Statistically significant support was not found for Hypothesis 1b, i.e., that 1NF users would find more anomalies than 3NF users. The results, however, were in the predicted direction.

Normalisation Level	Least-Squares Means	Significance of Pairwise Comparison with ¬NF	Significance of Pairwise Comparison with 1NF
¬NF	5.3462		
1NF	8.2917	0.0020	
3NF	7.2308	0.0396	0.2515

Table 2: Least squares means analysis for the number of anomalies found

5.3 Effects of Normalisation on the Efficiency of Error Finding

Hypothesis 2 predicted a direct relationship between the level of normalisation and the number of data anomalies found per query. The results of an ANOVA provided significant support for this hypothesis ($R^2 = 0.3089$, $F_{2,65} = 14.53$, $p = 0.0001$, two-tailed test). The least-squares means results (Table 3) confirm the relationship predicted by Hypothesis 2a, i.e., 1NF users found significantly more anomalies per query than ¬NF users. The results also support Hypothesis 2c, i.e., 3NF users found significantly more anomalies per query than ¬NF users. Statistically significant support was

not found for Hypothesis 2b, i.e., that 1NF users would find more anomalies per query than 3NF users. The results, however, were in the predicted direction.

Normalisation Level	Least-Squares Means	Significance of Pairwise Comparison with ¬NF	Significance of Pairwise Comparison with 1NF
¬NF	0.4838		
1NF	1.0056	0.0001	
3NF	0.8594	0.0003	0.1561

Table 3: Least squares means analysis for the number of anomalies found per query

5.4 Effects of Normalisation on the Number of Questions Attempted

Hypothesis 3 predicted a direct relationship between the level of normalisation and the number of questions attempted. The results of an ANOVA provided significant support for this hypothesis ($R^2 = 0.1313$, $F_{2,65} = 4.91$, $p = 0.0103$, two-tailed test). The least-squares means results (Table 4) support the relationship predicted by Hypothesis 3a, i.e., 1NF users attempted significantly more questions than ¬NF users. The results also support Hypothesis 3c, i.e., 3NF users attempted significantly more questions than ¬NF users. Statistically significant support was not found for Hypothesis 3b, i.e., that 1NF users would attempt more questions than 3NF users. The results, however, were in the predicted direction.

Normalisation Level	Least-Squares Means	Significance of Pairwise Comparison with ¬NF	Significance of Pairwise Comparison with 1NF
¬NF	4.7083		
1NF	6.0476	0.0036	
3NF	5.6522	0.0329	0.3805

Table 4: Least squares means analysis for the number of questions attempted

5.5 Summary of Experiment 1

The results from experiment 1 demonstrated that the normalisation level affected users' ability to retrieve data and thus to find data anomalies. The results showed that 1NF and 3NF data structures were more desirable for retrieving data than ¬NF data structures, i.e., 1NF and 3NF users found more anomalies, found more anomalies per query, and attempted more questions that ¬NF users. Although not statistically significant, results were consistent with hypotheses that 1NF data structures were more desirable for queries than 3NF data structures, i.e., 1NF users found more anomalies, found more anomalies per query, and attempted more questions than 3NF users. The non-significance of the results could, however, be a function of the experiment. Because the time available for the participants to perform the tasks in the experiment was limited, most participants only attempted to answer a small proportion of the questions. This resulted in participants answering few of the questions designed to test the differences between 1NF and 3NF data structures. These questions were further down the list and were only attempted by a few participants. To overcome this limitation and enable a better comparison between user performance with 1NF and 3NF data structures, a second experiment was conducted.

6. RESEARCH METHOD: EXPERIMENT 2

6.1 Research design, participants, and data collection

The purpose of experiment 2 was to test whether a 1NF data structure is more effective and efficient than a 3NF data structure for business end-users querying a database. This second experiment was similar to experiment 1.

The data structure (Appendix A), the scenario (Appendix B), the pretraining of participants, and the method of assignment to groups were identical to those experiment 1. There were three differences between the experiments. First, three levels of normalisation (¬NF, 1NF, and 3NF) were used in experiment 1 but only two levels of normalisation (1NF and 3NF) were used in experiment 2. Second, participants composed and executed SQL queries for 12 questions in experiment 1 whereas participants composed and executed SQL queries for 14 questions in experiment 2. These 14 questions resulted in queries (Appendix D) that differed substantially between the two levels of normalisation being tested. Third, there were 93 participants in experiment 2. These additional participants and the removal of a level of normalisation increased the number of participants in each group. While there were approximately 23 participants at each of the three normalisation levels in experiment 1, there were approximately 46 participants at each of the two normalisation levels in experiment 2.

6.2 Measures

The independent variable for all hypotheses was the level of normalisation (NL): 1NF or 3NF. The dependent variables were the number of questions attempted and the percent of correct anomalies found per query. Percent of anomalies found per query (as opposed to the number of anomalies found per query) was considered a more appropriate measure for this experiment because the number of anomalies per query ranged between 1 and 64. To determine the percent of correct anomalies found for each question, the anomalies found by the participants were divided by the number of anomalies seeded into the databases. The number of questions attempted was determined by examining the log files. Grade point average (GPA) was used as the covariate in each of the ANCOVAs.

7. RESULTS: EXPERIMENT 2

7.1 Summary Results

Table 5 summarises participant characteristics and performance by normalisation level.

	Normalisation Level	
	1NF	3NF
Grade Point Average (7-point scale, 7 highest)		
Mean	4.7111	4.9821
Standard deviation	0.8468	0.7648
Gender		
Number of males	28	29
Number of females	18	18
Total number of participants	46	47
Number of Requests Attempted		
Mean	8.5652	7.4468
Standard deviation	2.0939	2.1245
Percent of correct anomalies found/query		
Mean	77.5449	76.6953
Standard deviation	38.9135	40.2112

Table 5: Participant Characteristics and Summary Performance

7.2 Effects of Normalisation on Query Effectiveness

Hypothesis 1b and 2b predicted a direct relationship between the levels of normalisation and the data anomalies found. Comparing 1NF with 3NF, nested ANCOVA (GPA as covariate) results indicate that the percent of correct anomalies found was significantly associated with the normalisation level ($F_{11,725} = 2.00$, $p = 0.0262$, two-tailed test) (Table 6, Panel A). The means results (Table 5) confirm the relationship is in the direction predicted by hypothesis 1a, i.e., 1NF users identified a higher percentage of correct anomalies than 3NF users.

7.3 Effects of Normalisation on the Number of Questions Attempted

Hypothesis 3b predicted a direct relationship between the level of normalisation and the number of questions attempted. The results of a t-test ($t = 2.5564$, $p = 0.0122$, two-tailed test) (Table 6, Panel B) confirm the relationship is in the direction predicted by hypothesis 3b, i.e., 1NF users attempted significantly more questions than 3NF users.

Panel A: Effect on Percent of Correct Anomalies Detected

Source	R^2	df	Mean Square	F Value	Pr > F
Model	0.2106	25	24.66	7.74	0.0001
Error		725	92.43		
Query		13	1.55	12.15	0.0001
Normalisation Level (Query)		11	0.25	2.00	0.0262
GPA		1	3.05	23.94	0.0001

Panel B: Results of t-test with Number of Questions Completed

Normalisation Level	Mean	Std Dev	t value	p value
1NF	8.5652	2.0939	2.5564	0.0122
3NF	7.4468	2.1245		

Table 6: Main Model and Efficiency Effects as a Function of Normalisation Level (1NF, 3NF)

7.4 Post Hoc Analysis: Effects of Normalisation on Number and Type of Errors

Table 7 shows the total number of semantic errors participants made in retrieving the data anomalies. It also shows whether the errors were row, column, or aggregation errors.

Results of a nested ANCOVA (GPA as covariate) indicate that the number of errors was significantly associated with the normalisation level ($F_{11,725} = 16.43$, $p = 0.0001$, two-tail test) (Table 8, Panel A). The means results (Table 7) confirm that 1NF users made fewer errors composing queries to extract data anomalies than 3NF users.

Results of a nested ANCOVA (GPA as covariate) indicated that 3NF users on average made more column errors when composing their queries than 1NF users ($F_{11,725} = 7.47$, $p = 0.0001$, two-tail test) (Table 8, Panel B). The means results (Table 7) confirm the direction of the relationship. The results for row errors were also significant with 3NF users making more row errors when composing queries than 1NF users ($F_{11,725} = 27.19$, $p = 0.0001$, two-tail test) (Table 8, Panel C). The means results (Table 7) confirm the direction of the relationship. Although 3NF users made more errors than 1NF users, as expected, the result for aggregation errors was not statistically significant (Table 8, Panel D).

	Normalisation Level 1NF	3NF
Number of Errors		
Mean per attempted query	0.8655	1.5462
Standard deviation	1.4123	2.4918
Number of Column Errors		
Mean per attempted query	0.3020	0.4454
Standard deviation	0.7701	1.1094
Number of Row Errors		
Mean per attempted query	0.7232	0.9468
Standard deviation	2.2713	1.5379
Number of Aggregation Errors		
Mean per attempted query	0.1497	0.1541
Standard deviation	0.4285	0.4257
Number of Extraneous Data Anomalies Found		
Mean per attempted query	8.7056	12.8711
Standard deviation	24.5896	31.3460

Table 7: Effect of Normalisation Level on Performance

Panel A: Effect on Total Number of Errors

Source	R^2	df	Mean Square	F Value	Pr > F
Model	0.4442	25	54.75	23.18	0.0001
Error		725	2.36		
Query		13	71.23	30.16	0.0001
Normalisation level (Query)		11	38.81	16.43	0.0001
GPA		1	37.35	28.52	0.0001

Panel B: Effect on Column Errors

Source	R^2	df	Mean Square	F Value	Pr > F
Model	0.3405	25	9.19	14.97	0.0001
Error		725	0.61		
Query		13	13.50	21.99	0.0001
Normalisation level (Query)		11	4.59	7.47	0.0001
GPA		1	10.76	17.53	0.0001

Panel C: Effect on Row Errors

Source	R^2	df	Mean Square	F Value	Pr > F
Model	0.4785	25	21.07	26.61	0.0001
Error		725	0.79		
Query		13	22.19	28.03	0.0001
Normalisation level (Query)		11	21.53	27.19	0.0001
GPA		1	15.64	19.75	0.0001

(Continued next page)

Panel D: Effect on Aggregation Errors

Source	R^2	df	Mean Square	F Value	Pr > F
Model	0.1976	25	1.08	7.14	0.0001
Error		725	0.15		
Query		13	1.84	12.19	0.0001
Normalisation level (Query)		11	0.22	1.45	0.1461
GPA		1	0.94	6.24	0.0127

Table 8: Effect of Normalisation at 1NF and 3NF on Column, Row, and Aggregation Errors

Analysis of individual queries revealed that most of 3NF users' row errors were missing joins. These missing join clauses resulted in 3NF users detecting a larger number of records containing no anomalies, i.e., extraneous records. A nested ANCOVA (GPA as covariate) indicated that the number of extraneous records was marginally significant as a function of the normalisation level ($F_{11,725} = 1.62$, $p = 0.0898$, two-tail test) (Table 9). The means results (Table 7) confirm that 1NF users retrieved fewer extraneous records than 3NF users.

Source	R^2	df	Mean Square	F Value	Pr > F
Model	0.1296	25	3061.34	4.32	0.0001
Error		725	709.16		
Query		13	4516.66	6.37	0.0001
Normalisation level (Query)		11	1145.46	1.62	0.0898
GPA		1	9163.06	12.92	0.0003

Table 9: Effect of Normalisation at 1NF and 3NF on Number of Extraneous Data Anomalies

8. DISCUSSION AND CONCLUSIONS

Experimental results demonstrated that the level of normalisation affects users' ability to formulate database queries to find data anomalies. The results of experiment 1 showed that 1NF and 3NF data structures are more desirable for retrieving data than ¬NF data structures, i.e., 1NF and 3NF users found more anomalies, found more anomalies per query, and attempted more questions than ¬NF users. The results of experiment 1 did not, however, distinguish between user performance with 1NF and 3NF data structures. A likely cause of this outcome was that few participants attempted the questions for which the 1NF and 3NF queries differed. To distinguish between user querying performance with 1NF and 3NF data structures, participants in experiment 2 worked on questions with a higher proportion of queries that differed between the 1NF and 3NF versions.

Results of experiment 2 showed that 1NF data structures were better for querying than 3NF data structures, i.e., 1NF users found a significantly greater percent of the seeded data anomalies and attempted significantly more questions than 3NF users. This experiment also revealed that, when composing queries to extract data anomalies, 1NF users made significantly fewer errors than 3NF users. The higher error rate for 3NF users resulted in their queries retrieving more extraneous records that did not contain errors. Hence, compared with 1NF users, 3NF users, in addition to incurring greater information overload, would have more work to do to separate records containing data anomalies from those with no data anomalies.

Although higher levels of normalisation deter the introduction of data anomalies into a database, anomalies still accumulate. Data structures in 1NF provide business end-users the best likelihood of detecting such anomalies and only such anomalies through querying the data. Relational database theory, however, holds that higher levels of normalisation are desirable because they minimise data redundancy, which reduces the likelihood of data anomalies arising.

A solution to the problem of choosing between the better updates associated with high levels of normalisation and the better queries associated with 1NF is to have the best of both worlds by creating views for querying. The actual database can be normalised to 3NF or beyond, but views can be created that allow business end-users to query the data in 1NF. The impact of this conclusion is twofold. First, IS professionals need to assist or train business end-users in creating highly normalised data structures for updates and to make business end-users aware of the dangers of creating less normalised data structures. Second, IS professionals need to present business end-users with data views that minimise their querying difficulties. Based on this research, 1NF is the best data structure for business end-user database querying.

This study has several limitations. First, the study used students as participants. The participants had, however, received training in information technology (IT) and business related subjects, and thus are likely to be typical of business end-users that create their own systems. Second, some participants may require more training in SQL, particularly those not having a strong background in IT.

Future research could investigate the effects of normalisation on user satisfaction. Such research could provide insights into how satisfied users are with databases currently used by organisations. Low user satisfaction could be improved through training, use of different data structures, and access to views. Future research could also investigate the effect of normalisation on query errors. Such research could give insights into the number and type of errors users make when formulating queries using different data structures.

REFERENCES

BEYNON-DAVIES, P. (1992): Relational database design, Blackwell Scientific Publications, London.

BORTHICK, A.F., BOWEN, P.L. and SULLIVAN, M.C. (1998): Controlling JIT II: Making the system monitor itself, *Journal of Cost Management*, 12(4): 33-41.

BROOKSHIRE, R.G. (1993): A relational database primer, *Social Science Computer Review*, 11(2): 197-213.

CAMPBELL, D.J. (1988): Task complexity: A review and analysis, *Academy of Management Review*, 13(1): 40-52.

CHAN, H.C., WEI, K.K., and SIAU, K.L. (1993): User-database interface: the effect of abstraction level on query performance: a field experiment, *MISQ*, 17(4): 441-64.

CHAN, H.C., TAN, B.C.Y., and WEI, K.K. (1999): Three important determinants of user performance for database retrieval, *International Journal of Human Computer Studies*, 51: 895-918.

CODD, E.F. (1970): A relational model of data for large shared data banks, *Communications of the ACM*, 13(6): 377-87.

DATE, C.J. (1986): Relational Database: Selected Writings, Addison-Wesley Publishing Company, Reading, Massachusetts.

DATE, C.J. (1998): The birth of the relational model 3, *Intelligent Enterprize*, 1(4): 45-48.

DATE, C.J. (2000): An introduction to database systems, 7th Ed. Addison-Wesley Publishing Company, Reading, Massachusetts.

DAVIS, J.S. (1990): Experimental investigation of the utility of data structure and E-R diagrams in database query, *International Journal of Human Computer Studies*, 32:449-59

ELMASRI, R. and NAVATHE, S.B. (2001): Fundamentals of database systems, 3rd Ed, Addison-Wesley Publishing Company, Reading, Massachusetts.

HUH, Y.U., KELLER, F.R., REDMAN, T.C. and WATKINS, A.R. (1990): Data quality, information and software technology, 32(8): 559 - 565.

ICERMAN, R.C. and HILLISON, W.A. (1991): Disposition of audit-detected anomalies: Some evidence on evaluative materiality, *Auditing: A Journal of Practice and Theory*, Spring: 22-34.

JANSON, M.A. (1988): Data quality: The Achilles Heel of end-user computing, *Omega*, 16(5): 491-502.

KAOMEA, P. and PAGE, W. (1997): A flexible information manufacturing system to the generation of tailored information products, *Decision Support Systems*, 20(4): 345-355.

LAUDON, K.C. (1986): Data quality and due process in large interorganizational record systems, *Communications of the ACM*, 29(1): 4-11.

MEDAWAR, K. (1995): Database quality: A literature review of the past and a plan for the future, *Program*, 29(3): 257-272.

MOREY, R.C. (1982): Estimating and improving the quality of information in a MIS, *Communications of the ACM*, 25(5): 337-342.

REDMAN, T.C. (1995): Improve data quality for competitive advantage, *Sloan Management Review*, 36(2): 99-108.
SUH, K.S. and JENKINS, A.M. (1992): A comparison of linear keyword and restricted natural language database interfaces for novice users, *Information Systems Research*, 3(3): 252-72.

BIOGRAPHICAL NOTES

A.Faye Borthick, DBA, CMA, CPA, CISA, is Professor of Accountancy and Director of the Teaching and Learning with Technology Center, Georgia State University, Atlanta, GA, USA. For several decades, she has been working at the fault line of technology shifts in business and education. For courses in accounting and information systems, she has been developing new approaches to improving technology-enabled learning experiences, prompting students to solve problems with technology tools, and engaging students in their learning. She has been teaching completely online courses since 1997.

Paul L. Bowen, PhD, is Senior Lecturer in Information Systems, University of Queensland, Brisbane, Australia, where he teaches information systems and auditing. His primary research includes developing analytical models of data quality, measuring the impact of data errors on decision quality, and applying artificial intelligence techniques to the identification and classification of data errors. He also has research interests in database design, internal control, and software reliability. He has been a systems analyst and project manager at Oak Ridge National Laboratory and has taught at The University of Tennessee and Auburn University. http://www.commerce.uq.edu/staff/bowen.phtml

Fiona H. Rohde, PhD, is Senior Lecturer in Information Systems, University of Queensland, Brisbane, Australia, where she teaches Information Management and Systems Analysis and Design. Her research interests include the areas of outsourcing, data and information management, e-commerce and their affect on audit and accounting techniques. Fiona worked for KPMG Peat Marwick in the Computer Audit Division assisting in team audits in mining and manufacturing firms before joining the school approximately 10 years ago.

Meng R. Liu, MInfSys, is a Lead Business Analyst working for Optus, Australia. He has six years of IT and Management Consulting experience (Ernst & Young Consulting & The LiTMUS Group) where he worked on consulting engagements in a variety of industries including utilities, telecommunications, manufacturing, banking, finance and insurance. Meng's key areas of expertise are in Business Analysis, Data Warehousing, EIS, Business Process Reengineering, Business Performance Management, and Business Intelligence.

APPENDIX A
DATA STRUCTURES
Non-Normalised ((NF)

Attribute Name	Type	Description	Comments
ITEMA Table			
itemno	Char(6)	Item number of item	
vno	Char(5)	Vendor number	
vname	Char(30)	Name of the vendor	
idesc	Char(30)	Description of item	
terms	Char(5)	Terms of payment	(Violates atomicity - Violates 1NF)
ctqtyoh	Number	Current quantity on hand	
prodjno1	Char(5)	Production job number1	(Repeating fields - Violates 1NF)
issdate1	Date	Date the item1 was issued	
qtyiss1	Number	Quantity of the item issued	
qtydef1	Number	Quantity of the defective item 1	
prodvar1	Number	Production time variance 1. E.g. 1.10 indicates production required 110% of standard time, i.e., a 10% unfavourable variance.	
prodjno2	Char(5)	Production job number 2	
issdate2	Date	Date the item 2 was issued	
qtyiss2	Number	Quantity of the item issued	
qtydef2	Number	Quantity of the defective items 2	
prodvar2	Number	Production time variance 2.	
prodjno3	Char(5)	Production job number 3	
issdate3	Date	Date the item 3 was issued	
qtyiss3	Number	Quantity of the item 3 issued	
qtydef3	Number	Quantity of the defective items 3	
prodvar3	Number	Production time variance 3.	
recrepno1	Char(6)	No of the first report	(Repeating fields - Violates 1NF)
recdate1	Date	Date shipment 1 received	
qtyrec1	Number	Quantity of item 1 received	
unitcost1	Number	Cost per unit 1	
paydate1	Date	Date payment 1 made	(payment corresponds to receiving report)
payamt1	Number	The amount 1 paid	
recrepno2	Char(6)	No of the second report	
recdate2	Date	Date shipment 2 received	
qtyrec2	Number	Quantity of item 2 received	
unitcost2	Number	Cost per unit 2	
paydate2	Date	Date payment 2 made	(payment corresponds to receiving report)
payamt2	Number	The amount 2 paid	

First Normal Form (1NF)

Attribute Name	Type	Description
ITEMB Table		
itemno	Char(6)	Item number of item
vno	Char(5)	Vendor number
vname	Char(30)	Name of the vendor
idesc	Char(30)	Description of item
termdisc	Number	Terms of payment - discount percent
termdays	Number	Terms of payment - number of days discount applies
cqtyoh	Number	Current quantity on hand
ISSUESB Table		
prodjno	Char(5)	Production job number
itemno	Char(6)	Item number of item
issdate	Date	Date the item was issued
qtyiss	Number	Quantity of the item issued
qtydef	Number	Quantity of the defective items
prodvar	Number	Effect on production time, i.e., production time variance. For example 1.10 indicates production required 110% of standard time, i.e., a 10%unfavourable variance.
RECEIPTSB Table		
recrepno	Char(6)	The number of the receiving report
itemno	Char(6)	Item number of item
recdate	Date	Date shipment received
qtyrec	Number	Quantity received
unitcost	Number	Cost per unit
paydate	Date	Date payment was made (the payment corresponds to the receiving report)
payamt	Number	The amount paid

Third Normal Form (3NF)

Attribute Name	Type	Description
ITEMC Table		
itemno	Char(6)	Item number of item
idesc	Char(30)	Description of item
vno	Char(5)	Vendor number
cqtyoh	Number	Current quantity on hand
VENDORC Table		
vno	Char(5)	Vendor number
vname	Char(30)	Vendor name
termdisc	Number	Terms of payment - discount percent
termdays	Number	Terms of payment - number of days discount applies
ISSUESC Table		
prodjno	Char(5)	Production job number
itemno	Char(6)	Item number of item
issdate	Date	Date the item was issued
qtyiss	Number	Quantity of the item issued
qtydef	Number	Quantity of the defective items
PRODUCTIONC Table		
prodjno	Char(5)	Production job number
itemno	Char(6)	Item number of item
prodvar	Number	Effect on production time, i.e., production time variance. For example 1.10 indicates production required 110% of standard time, i.e., a 10% unfavourable variance.
RECEIPTSC Table		
recrepno	Char(6)	The number of the receiving report
itemno	Char(6)	Item number of item
recdate	Date	Date shipment received
qtyrec	Number	Quantity received
unitcost	Number	Cost per unit
PAYMENTSC Table		
recrepno	Char(6)	The number of the receiving report
itemno	Char(6)	Item number of item
paydate	Date	Date payment was made (the payment corresponds to the receiving report)
payamt	Number	The amount paid

APPENDIX B

Instructions - ¬NF, 1NF and 3NF

Purpose

Some of the purposes of this exercise are to:

1. Test how well you can use SQL to search for data anomalies within a database.
2. Allow you to relate input controls to specific data anomalies by searching for data anomalies that input controls should have prevented.

Task Overview

Task 1 - Read the Scenario and Chris's Questions. (5 mins.)

Task 2 - Record your session (minimal time).

Task 3 - Locate anomalies and problems Chris (your boss) head of the internal audit department, specifically asked you to look for. (40 mins.)

Task 4 - Report records that contain anomalies or problems Chris specifically asked you to look for. (10 mins.)

Task 5 - Use your ingenuity to locate anomalies and problems not specifically requested by Chris. (40 mins.)

Task 6 - Report records that contain anomalies or problems not specifically requested by Chris. (10 mins.)

Task 7 - Transmit log files. (minimal time)

Task 8 - Complete the survey. (5 mins.)

Scenario

Background

Comfortable Furniture Limited manufactures household and office furniture for distribution throughout the world. The company operates from 9:00 am to 5:00 pm, Monday through to Saturday. Comfortable Furniture adopted the Just-In-Time (JIT) II method a little over a year ago for inventory records. JIT II involves the use of contractual agreements between an organisation and its suppliers where the suppliers assume direct responsibility for entire categories of the organisation's inventory. Vendors: (1) provide the required items on a just-in-time basis for production schedules; (2) provide these items at favourable, if not preferential prices; and (3) over the long term, make innovations in their products, production, and pricing to match the organisation's requirements. In JIT II situations, vendor representatives often occupy offices in the organisation's facilities. The organisation grants the vendor's representatives access to the organisation's data and freedom to inspect physical inventory. The vendor representatives, rather than personnel in the organisation's purchasing department, place orders for the materials needed for the organisation's production runs.

JIT II reduces ordering costs, delivery times, handling costs, and inventory holding expenses. For these systems to yield the intended benefits, suppliers must not abuse their direct ordering capabilities. Converting to JIT II means many of the internal controls for traditional purchasing procedures are eliminated. Management expects the internal auditors to query the information systems to determine that JIT II vendor relationships meet internal control objectives.

Task

On 2 July 1995, Chris Kaniuk, the manager of the internal audit group (your boss), calls you into her office. She wants you to scrutinise the JIT II database using SQL SELECT queries to locate any data anomalies before Deer, Price, and Persnickety, the external auditors, begin their investigation of the system. Furthermore, Ian McMurdy, president of Comfortable Furniture, asked Chris to look for problems or potential problems with the JIT II system. He particularly wants to know about potential overpayments to vendors and whether the overpayments arose from anomalies or irregularities. Because you must investigate the data anyway, Chris just delegated you this task as well. After completing your investigation, prepare a memo (in bullet format) to Chris pointing out the anomalies and problems you found in the JIT II database.

Part 1 - Answer Chris's questions - EXPERIMENT 1

1. Items with no item description.
2. Negative quantities issued for items.
3. Production variances less than 0.8 or greater than 1.5.
4. (a) Term discount rates greater than 5% and
 (b) Term discount days greater than 45 days.
5. Same item description for different item numbers.
6 Items that have current quantity on hand less than the average quantity issued.
7. Items with unit prices that have differences greater than 50% between different receipts.
8. Discount [or possibly even an incorrect discount %] taken after the discount period.
9. Production variance greater than 1.1 and quantity defective = 0
10. Payment date less than receipt date plus half the number of term days
11. Items that have total receipts greater than 10 times the current quantity on hand.
12. Vendor with the highest percentage of defective items

Part 1 - Answer Chris's questions - EXPERIMENT 2

1. List the item numbers, issue date, and quantity issued for issues with negative quantity issued.
2. List vendors (numbers and names) and the items (item numbers and descriptions) they supply for vendors from whom we receive a 5% or greater discount.
3. List vendors (numbers and names) and the total number of items each vendor supplies.
4. List the item numbers for items with no item descriptions. Check for both missing descriptions and descriptions with blank spaces.
5. Management wants an idea of which vendors are supplying poor quality products. List vendor names and the ratio of the sum of the quantity of defective items to the sum of quantity of the items issued. Sort the vendors by highest to lowest defective ratio.
6. Management is very concerned about cash management and wants to know the payments that we have made earlier than necessary. List the vendor names, item numbers, receiving report numbers, receipt date, payment date, and discount days for transactions where the payment was made when less than half the discount days had elapsed.
7. Management wants an estimate of the current accounts payable. Calculate the total gross amount payable for materials we have not paid for yet.

8. List items for which the current quantity on hand is more than 50 but there is no usable item description, i.e., the item description is either blank or less than 3 characters. Include the vendor names associated with each item.
9. By vendor, management wants to know the amount of discounts we have lost by not paying within the discount period. List vendor names, each vendors' discount days, and the sum of the amount of the discounts lost. Each discount lost can be calculated as the product of the discount percentage times the gross dollar amount received. The gross dollar amount received is the product of the quantity received times the unit cost.
10. JIT II arrangements typically involve obtaining particular items from only one vendor, i.e., each vendor supplies all of the organisation's requirements for its designated set of items. Management wants to know if the manufacturing staff have managed to arrange to obtain the same item from more than one vendor. List pairs of vendor names and pairs of item numbers and descriptions for items supplied by different vendors but with the same item descriptions.
11. Management wants to look for items that may be overstocked. List the vendor names, item numbers and descriptions, current quantity on hand, and the average daily issue since 1 January 2000 for items having current quantity on hand greater than 10 times the average daily issue for that item. Recall that the average daily issue of an item is the total amount issued for the period divided by the number of days in the period.
12. Management is concerned that manufacturing is not recording information about production variances. List production job numbers, item numbers and descriptions, and issue dates for which production variances were missing or not recorded.
13. List items for which the value of the current quantity on hand is more than $1,000 (based on the most recent unit cost for a receipt of that item) but there is no usable item description, i.e., the item description is either blank or less than 3 characters. Include the vendor names associated with each item as well as the current quantity on hand of each item and the estimate of the value of the current quantity on hand of those items.
14. Management wants to identify vendors with consistently high quality items. List vendor names and average defective rates (total quantity defective divided by total quantity issued) for vendors which supply items for which none of the items the vendor supplies has a deficiency rate greater than 5%.

APPENDIX C
Questions and Queries: Experiment 1

Question*	¬NF Query	1NF Query	3NF Query
1. Items with no item description	Select distinct itemno From itema Where idesc= ' ' or idesc is null;	Select itemno From itemb Where idesc= ' ' or idesc is null;	Select itemno From itemc Where idesc= ' ' or idesc is null;
2. Negative quantities issued for items	Select itemno, issdate1, qtyiss1, issdate2, qtyiss2, issdate3, qtyiss3 From itema Where qtyiss1 < 0 or qtyiss2 < 0 or qtyiss3 < 0;	Select itemno, issdate, qtyiss From issuesb Where qtyiss < 0;	Select itemno, issdate, qtyiss From issuesc Where qtyiss < 0;
3. Production variances less than 0.8 or greater than 1.5	Select itemno, prodjno1, prodvar1, prodjno2, prodvar2, prodjno3, prodvar3 From itema Where (prodvar1 > 1.5 or prodvar1 < 0.8) or (prodvar2 > 1.5 or prodvar2 < 0.8) or (prodvar3 > 1.5 or prodvar3 < 0.8);	Select itemno, prodjno, prodvar From issuesb Where prodvar > 1.5 or prodvar < 0.8;	Select itemno, prodjno, prodvar From productionc Where prodvar > 1.5 or prodvar < 0.8;
4. Term discount rates greater than 5% or term discount days greater than 45 days.	Select distinct vno, vname, terms From itema Where to_number (substr(terms,1,instr(terms,'/')-1))>5 or to_number (substr(terms,3,instr(terms,'/')+2))>45;	Select vno, vname, termdisc, termdays From itemb Where termdisc > 5 or termdays > 45;	Select vno, vname, termdisc, termdays From vendorc Where termdisc > 5 or termdays > 45;
5. Same item description for different item numbers	Select distinct a.itemno, a.idesc From itema a, itema b Where a.itemno <> b.itemno and a.idesc = b.idesc;	Select a.itemno, a.idesc From itemb a, itemb b Where a.itemno <> b.itemno and a.idesc = b.idesc;	Select a.itemno, a.idesc From itemc a, itemc b Where a.itemno <> b.itemno and a.idesc = b.idesc;
6. Items that have current quantity on hand less than the average quantity issued.	Select distinct a.itemno From itema a Where a.cqtyoh < (select avg((b.qtyiss1+b.qtyiss2+b.qtyiss3)/3) from itema b where b.itemno=a.itemno group by b.itemno);	Select itemb.itemno, avg(qtyiss) From itemb, issuesb Where itemb.itemno=issuesb.itemno and cqtyoh < (select avg(qtyiss) from issuesb where issuesb.itemno=itemb.itemno) group by itemb.itemno;	Select itemc.itemno, avg(qtyiss) From itemc, issuesc Where itemc.itemno=issuesc.itemno and cqtyoh < (select avg(qtyiss) from issuesc where issuesc.itemno=itemc.itemno) group by itemc.itemno;

Question*	¬NF Query	1NF Query	3NF Query
8. Discount [or possibly even an incorrect discount percent] taken after the discount period	Select itemno, recrepno1, recdate1, paydate1, payamt1, recrepno2, recdate2, paydate2, payamt2 From itema Where(paydate1 >recdate1 +(to_number (substr(terms,3,instr(terms,'/')+2))) and (payamt1 < unitcost1*qtyrec1) or (paydate2 > recdate2 + (to_number (substr(terms,3,instr(terms,'/')+2))) and (payamt2 < unitcost2 * qtyrec2);	Select receiptsb.itemno, receiptsb.recrepno, receiptsb.recdate, receiptsb.paydate, receiptsb.payamt From receiptsb, itemb Where itemb.itemno=receiptsb.itemno and paydate > recdate + termdays and payamt < unitcost*qtyrec;	Select receiptsc.itemno, receiptsc.recrepno, receiptsc.recdate, paymentsc.paydate, paymentsc.payamt From receiptsc, vendorc, paymentsc, itemc Where itemc.itemno=receiptsc.itemno and receiptsc.recrepno=paymentsc.recrepno and itemc.vno=vendorc.vno and paydate > recdate + termdays and payamt < unitcost*qtyrec;
9. Production variance greater than 1.1 and quantity defective = 0	Select distinct itemno From itema Where ((prodvar1 >1.1) and (qtydef1 =0)) or ((prodvar2 > 1.1) and (qtydef2 = 0)) or ((prodvar3 > 1.1) and (qtydef3 =0));	Select itemb.itemno From itemb, issuesb Where itemb.itemno=issuesb.itemno and prodvar > 1.1 and qtydef = 0;	Select issuesc.itemno From issuesc, productionc Where issuesc.prodjno=productionc.prodjno and prodvar > 1.1 and qtydef = 0;
10. Payment date less than receipt date plus half the number of term days	Select itemno, recrepno1, recrepno2 From itema Where paydate1 < (recdate1+(0.5*(to_number (substr(terms,3,instr(terms,'/')+2))))) or paydate2 < (recdate2+(0.5*(to_number (substr(terms,3,instr(terms,'/')+2)))));	Select itemb.itemno, recrepno From itemb, receiptsb Where itemb.itemno=receiptsb.itemno and paydate < (recdate+(0.5*termdays));	Select itemc.itemno, receiptsc.recrepno From itemc, vendorc, receiptsc, paymentsc Where itemc.vno=vendorc.vno and receiptsc.itemno=itemc.itemno and paymentsc.recrepno=receiptsc.recrepno and paydate < (recdate+(0.5*termdays));
12. Vendor with the highest percentage of defective items	Select itemno, (sum(qtydef1+qtydef2+qtydef3)/ sum(qtyiss1+qtyiss2+qtyiss3)) From itema Group by itemno having (sum(qtydef1+qtydef2+qtydef3)/ sum(qtyiss1+qtyiss2+qtyiss3)) = (select max (sum(qtydef1+qtydef2+qtydef3) /sum(qtyiss1+qtyiss2+qtyiss3)) from itema group by itemno);	Select itemno, (sum(qtydef)/sum(qtyiss)) From issuesb Group by itemno having (sum(qtydef)/sum(qtyiss)) = (select max(sum(qtydef)/sum(qtyiss)) from issuesb group by itemno);	Select itemno, (sum(qtydef)/sum(qtyiss)) From issuesc Group by itemno having (sum(qtydef)/sum(qtyiss)) = (select max(sum(qtydef)/sum(qtyiss)) from issuesc group by itemno);

* Questions 7 and 11 were not attempted by any participants and thus have been deleted from the sample queries.

APPENDIX D
Questions and Queries: Experiment 2

Question*	1NF Query	3NF Query
1. List the item numbers, issue date, and quantity issued for issues with negative quantity issued.	Select itemno, issdate, qtyiss From issuesb Where qtyiss < 0.0;	Select itemno, issdate, qtyiss From issuesc Where qtyiss < 0.0;
2. List vendors (numbers and names) and the items (item numbers and descriptions) they supply for vendors from whom we receive a 5% or greater discount.	Select vno, vname, itemno, idesc From itemb Where termdisc ≥ 0.05;	Select vendorc.vno, vname, itemno, idesc From vendorc, itemc Where vendorc.vno = itemc.vno and termdisc ≥ 0.05;
3. List vendors (numbers and names) and the total number of items each vendor supplies.	Select vno, vname, count(*) From itemb Group by vno, vname;	Select vendorc.vno, vname, count(*) From vendorc, itemc Where vendorc.vno = itemc.vno Group by vendorc.vno, vname;
4. List the item numbers for items with no item descriptions. Check for both missing descriptions and descriptions with blank spaces.	Select itemno From itemb Where idesc = ' ' or idesc is null;	Select itemno From itemc Where idesc = ' ' or idesc is null;
5. Management wants an idea of which vendors are supplying poor quality products. List vendor names and the ratio of the sum of the quantity of defective items to the sum of quantity of the items issued. Sort the vendors by highest to lowest defective ratio.	Select vname, sum(qtydef)/sum(qtyiss) From itemb, issuesb Where itemb.itemno = issuesb.itemno Group By vname Order By 2 Desc;	Select vname, sum(qtydef)/sum(qtyiss) From vendorc, itemc, issuesc Where vendorc.vno = itemc.vno and itemc.itemno = issuesc.itemno Group By vname Order By 2 Desc;
6. Management is very concerned about cash management and wants to know the payments that we have made earlier than necessary. List the vendor names, item numbers, receiving report numbers, receipt date, payment date, and discount days for transactions where the payment was made when less than half the discount days had elapsed.	Select vname, itemb.itemno, recrepno, recdate, paydate, termdays From itemb, receiptsb Where itemb.itemno = receiptsb.itemno and paydate < recdate + (0.5 * termdays);	Select vname, itemc.itemno, receiptsc.recrepno, recdate, paydate, termdays From vendorc, itemc, receiptsc, paymentsc Where vendorc.vno = itemc.vno and itemc.itemno = receiptsc.itemno and receiptsc.recrepno = paymentsc.recrepno and receiptsc.itemno = paymentsc.itemno and paydate < recdate + (0.5 * termdays);

Question*	1NF Query	3NF Query
7. Management wants an estimate of the current accounts payable. Calculate the total gross amount payable for materials we have not paid for yet.	Select sum(qtyrec*unitcost) From receiptsb Where paydate IS NULL;	Select sum(qtyrec*unitcost) From receiptsc, paymentsc Where receiptsc.recrepno = paymentsc.recrepno and receiptsc.itemno = paymentsc.itemno and paydate is NULL;
8. List items for which the current quantity on hand is more than 50 but there is no usable item description, i.e., the item description is either blank or less than 3 characters. Include the vendor names associated with each item.	Select vno, vname, itemno, cqtyoh From itemb Where cqtyoh > 50 and (idesc is null or length(idesc) < 3);	Select vendorc.vno, vname, itemno, cqtyoh From vendorc, itemc Where vendorc.vno = itemc.vno and cqtyoh > 50 and (idesc is null or length(idesc) < 3);
9. By vendor, management wants to know the amount of discounts we have lost by not paying within the discount period. List vendor names, each vendors' discount days, and the sum of the amount of the discounts lost. Each discount lost can be calculated as the product of the discount percentage times the gross dollar amount received. The gross dollar amount received is the product of the quantity received times the unit cost.	Select vname, termdays, sum(termdisc * qtyrec * unitcost) From itemb, receiptsb Where itemb.itemno = receiptsb.itemno and paydate - recdate > termdays Group By vname, termdays;	Select vname, termdays, sum(termdisc * qtyrec * unitcost) From vendorc, itemc, receiptsc, paymentsc Where vendorc.vno = itemc.vno and itemc.itemno = receiptsc.itemno and receiptsc.recrepno = paymentsc.recrepno and receiptsc.item_no = paymentsc.item.no and paydate - recdate > termdays Group By vname, termdays;
10. JIT II arrangements typically involve obtaining particular items from only one vendor, i.e., each vendor supplies all of the organisation's requirements for their designated set of items. Management wants to know if the manufacturing staff have managed to arrange to obtain the same item from more than one vendor. List the pairs of vendor names and pairs of item numbers and descriptions for items supplied by different vendors but with the same item descriptions.	Select a.vname, b.vname, a.itemno, b.itemno, a.idesc, b.idesc From itemb a, itemb b Where a.itemno <> b.itemno and a.idesc = b.idesc;	Create View sameidesc As Select a.vno avno, b.vno bvno, a.itemno aitemno, b.itemno bitemno, a.idesc aidesc, b.idesc bidesc From itemc a, itemc b Where a.itemno <> b.itemno and a.idesc = b.idesc; Select c.vname, d.vname, aitemno, bitemno, aidesc, bidesc From vendorc c, vendorc d, sameidesc Where c.vno = avno and d.vno = bvno;

* Questions 11, 12, 13, and 14 were not attempted by any participants and have been omitted from the query list.

The Code Red Epidemic: a Case Study

John Lamp
Faculty of Business and Law, Deakin University, Australia
Email: John.Lamp@deakin.edu.au

An analysis of log files from an immune World Wide Web server was used to discover the patterns of infection from the Code Red worm variants. Analogies are drawn to biological systems. The need for protection is commented on.
Keywords: EK10 Computer Viruses, EL08 IS Risk Management

INTRODUCTION

Computer viruses have become a fact of life. Once a rarity, the increasing use of networks and network applications, such as email, has resulted in their rapid dispersion after creation. Their increasing prevalence is providing data that can be analysed to obtain information and devise models on rates of infection, spatial dispersion and other aspects.

THE CODE RED WORM

The first variant of the Code Red worm (CODERED.A) was discovered on the 19th July 2001. Subsequently two variations of the worm were discovered, CODERED.B on 31st July 2001, and CODERED.C on 6th August 2001. The worm specifically attacks web servers running Microsoft's Internet Information Server (IIS) on Windows NT 4, 2000 or XP Beta platform. The worm exploits an unchecked buffer in the Internet Data Administration area of IIS, contained in idq.dll. It creates a buffer overflow and gains complete control over the server. The actions the worm then takes is dependent on the variant of the worm.

Variant A is relatively benign. If the system date is before the 20th of the month, it generates random IP addresses and sends copies of itself to other web servers. If the system date is between the 20th and 28th, it executes a distributed denial of service (DDoS) attack on 198.137.240.91 (www1.whitehouse.gov). This server has now been moved to a different IP address. It also checks for a file called C:\NOTWORM. If this file exists, the worm goes dormant. It also causes the server on which it resides to display the message "Welcome to http://www.worm.com! Hacked By Chinese!" when accessed. Typically you see something like the following line in the log file, where [240*"N"] is a string of 240 "N"s.

```
GET /default.ida?[240*"N"]%u9090%u6858%ucbd3%u7801%u9090
```

Variant B fixes a bug in variant A, which caused variant A to generate the same random addresses. It displays the hacked message only if the system language is not English.

Variant C can be identified in the log files as it uses a string of "X"s rather than "N"s. Variant C

Copyright© 2001, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the JRPIT copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Australian Computer Society Inc.

Manuscript received: August 2001
Editor: John Roddick

seems to work properly only on Windows 2000 machines. It causes NT 4 machines to crash. Variant C spawns either 300 or 600 threads to infect other machines compared with the earlier variants' 99 threads. Variant C is also more malicious in that it additionally installs a trojan version of Explorer and disables system file protection by modifying the registry. It also maps the C: and D: drive to web accessible directories /C and /D, thus making any file on the machine web accessible. Yet another good reason to have directory browsing turned off as default! It does not have a dormant period, nor undertake a DDoS attack. (Trend, 2001a; Trend, 2001b; Trend, 2001c; eEye, 2001; Microsoft, 2001)

LOG FILE ANALYSIS

The machine that supplied the case study data is running the Sambar Web Server (Sambar, 2001) on a Windows 2000 Professional platform. It is not behind any firewall. The log file accumulated has the format in Table 1.

remotehost	Remote hostname or IP address number if DNS is not enabled/available.
rfc931	The remote login name of the user. (This is not implemented by the Sambar Server).
authuser	The username of the authenticated user. This is available when using password protected WWW pages.
[date]	Date and time of the request.
"request"	The HTTP request line as it came from the client.
status	The HTTP response code returned to the client. Indicates whether or not the file was successfully retrieved, and if not, what error message was returned.
bytes	The number of bytes transferred. If the status is 200 and bytes are 0, the dynamic page size could not be determined.
msec "referer"	The time in milliseconds that it took the server to respond to the request. The url the client was on before requesting this url.
"agent"	The browser the client is using.

Table 1: Sambar Server log file format (Sambar, 2001)

A CODERED.A infection attempt was first logged on the case study machine on 20 July 2001. It later re-appeared on 30 July 2001, following which it has become a permanent feature, with CODERED.C dominant at around 150 attempts per day. In total 2,539 incidents have been recorded up to and including 20th August 2001. The grep utility was used to extract log file lines caused by virus infection attempts. These records were imported into an Access database and reformatted so that a cross tabulation of date against variant of virus could be run. The results were then graphed using Excel. (Figure 1).

The log file for 20th of August 2001 has no entries for CODERED.A or .B, indicating it has entered its dormant period. CODERED.C is still being logged. Its trend is downward, but slower than that of variant A/B.

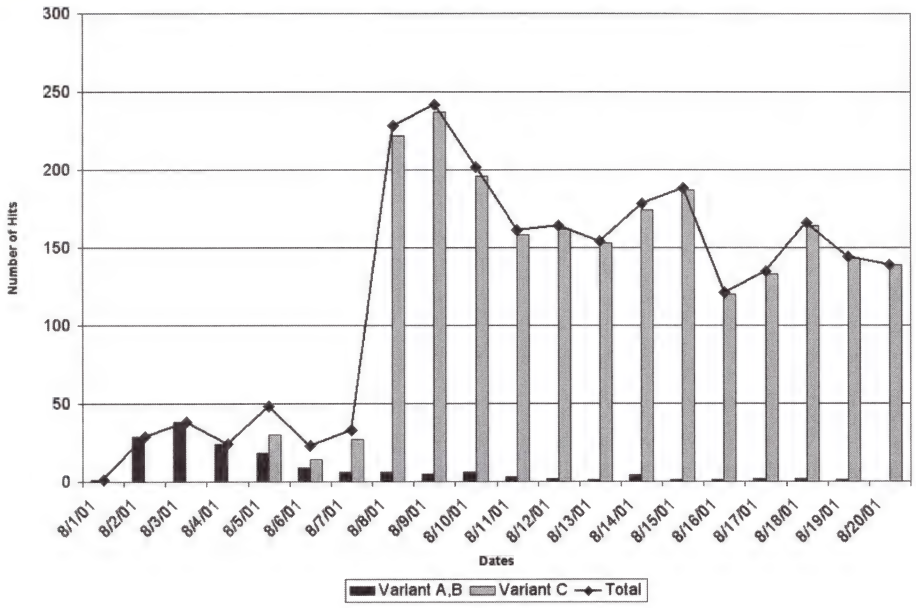


Figure 1: Code Red Infection Attempt Pattern – August

BIOLOGICAL ANALOGUES

Information technology is an area rich in metaphor, and the terms used to describe biological epidemics are often used loosely to describe the actions of computer viruses. (eg Kephart et al, 1993, 1997) In particular, epidemiological terms, such as infectives that introduce the disease, susceptibles that are able to be infected, and immunes that cannot be infected, are often used. From the quantitative theory behind biological epidemic analysis, graphs may be constructed, which describe the progress of the epidemic. (Poole, 1974) (Figure 2).

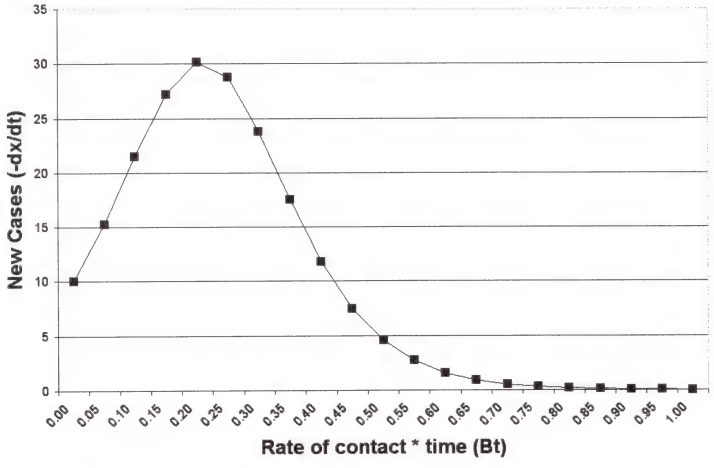


Figure 2: Deterministic epidemic curve for a simple epidemic (after Bailey, 1957)

Biological epidemic analysis is based on the number of new cases reported, and the requirement to report is, most often, a statutory one placed on the health profession. In this analysis, what was actually recorded was unsuccessful infection attempts. There is no parallel to this in classical epidemiology – it is not really possible to say, "That breath contained a flu virus particle that did not infect me!" Intuitively it might seem that the number of infection attempts should be related to the number of infectives in the population, but this requires a number of assumptions on how infectives are actually attempting to spread their infection, and needs closer examination. Certainly the literature in this area has a large number of mathematical models, which could be useful in understanding and predicting the impact of computer viruses on the normal operation of systems. Equally, there are a number of areas, such as modelling the spatial spread of biological epidemics, which have not been able to be undertaken effectively, but may be able to be addressed in the information technology domain. These could then be examined in the biological domain.

CONCLUSION

The increased prevalence of viruses and the ability to log attempts by viruses to attach machines is resulting in a greater amount of data, which could form the basis of new methods for modelling their distribution. Existing models of biological epidemics may be applicable to epidemics of computer viruses. In this specific case, large numbers of infection attempts are still being logged despite a corrective patch being issued on 18th June 2001, prior to the discovery of the virus in the wild. (Microsoft, 2001).

REFERENCES

BAILEY, N. J. T. (1957): *The Mathematical Theory of Epidemics,* Charles Griffin & Co, London

eEYE DIGITAL SECURITY (2001): AD20010618 *All versions of Microsoft Internet Information Services Remote buffer overflow (SYSTEM Level Access)* [Online]
Available: http://www.eeye.com/html/Research/Advisories/AD20010618.html
Last Accessed: 2001-08-20

KEPHART, J. O., CHESS, D. M. and WHITE, S R. (1993) *"Computers and Epidemiology" IEEE Spectrum* [Online]
Available: http://www.research.ibm.com/antivirus/SciPapers/Kephart/Spectrum/Spectrum.html
Last Accessed: 2001-08-20

KEPHART, J. O., SORKIN, G B. CHESS, D M. and WHITE, S R. (1997) *"Fighting Computer Viruses"* Scientific American [Online] Available: http://www.sciam.com/1197issue/1197kephart.html
Last Accessed: 2001-08-20

MICROSOFT (2001) MS01-033 *Unchecked Buffer in Index Server ISAPI Extension Could Enable Web Server Compromise* [Online]
Available: http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-033.asp
Last Accessed: 2001-08-20

POOLE, R. (1974) *An Introduction to Quantitative Ecology,* McGraw-Hill, New York

SAMBAR, T. (2001) Sambar Server [Online]
Available: http://www.sambar.com/ Last Accessed: 2001-08-20

TREND (2001a) CODERED.A [Online] Available:
http://www.antivirus.com/vinfo/virusencyclo/default5.asp?VName=CODERED.A Last Accessed: 2001-08-20

TREND (2001b) CODERED.B [Online] Available:
http://www.antivirus.com/vinfo/virusencyclo/default5.asp?VName=CODERED.B Last Accessed: 2001-08-20

TREND (2001c) CODERED.C [Online] Available:
http://www.antivirus.com/vinfo/virusencyclo/default5.asp?VName=CODERED.C Last Accessed: 2001-08-20

BIOGRAPHICAL NOTE

John Lamp is a lecturer in the School of Management Information Systems at the Waurn Ponds campus of Deakin University. He originally graduated as a zoologist from the University of Tasmania. He worked in the public sector in a variety of programme and support positions covering information technology, change management and strategic management. He commenced an academic career at the University of Tasmania in 1995. In 1998 he joined Deakin University. John's interests include systems development methodologies, and project management.

LAVINGTON, S. (2000): The Pegasus Story: A history of a vintage British computer. Science Museum, London, 58pp., (pound sterling) 6.95 (Paper Back)

This is an excellent description of how PEGASUS was conceived, designed, built and used. It is well set out and compares and contrasts PEGASUS with other similar machines which were produced in the same time frame. There are five chapters as follows:-

1. "What is PEGASUS?" This is a very short three pages, "scene setting" which gives a useful background of the technogogy available in the early 1950s and so it explains why it came to be designed as it was.
2. "The birth of PEGASUS." This elaborates on the design objectives and emphasises the most important point that it was intended to be easy for programmers to use it to achieve high-speed operation with no thought needed to arrange the instructions in the memory. This is in contrast to the English Electric DEUCE, which had a faster clock speed (see Table 5.1) but the Deuce needed the programmer to arrange the instructions on the storage drum optimally a tedious task which can also be data dependent. (Programmers' time is very valuable!) The PEGASUS was also cheaper than the DEUCE – see Table 5.2 – and this combined with ease of programming explains why Ferrantis sold more than English Electric sold Deuces.
3. "Technical description of the Ferranti PEGASUS." Included here is a list of the machine's orders/instructions; so even today a really keen student could write a program using this information. Furthermore, a PEGASUS simulator, which runs on a PC, has been written by members of the Computer Conservation Society and it is available for downloading from an internet web site, whose address is given on page iv.
4. "PEGASUS in action." This is a chapter which this reviewer considers to be much too short to do justice to all the problems which were solved using PEGASUSs' during the late 1950s and 1960s.
5. "PEGASUS's place in history." Here the Author compares PEGASUS to several other contemporary and later computers (Table 5.1) and explains that the machine (serial No. 25) – originally exported to Skandia, in Sweden, has gone full circle and is now back in the UK and is fully operational at the Science Museum in London. This restoration has been no mean feat considering the fact that most of the electronic components are now obsolete.

My main criticism of the book is that it contains no references but refers the reader to a list of 70 which are held at a .uk Web site. I feel very strongly that any book of this type should be self-contained and should not rely on it's readers having access to the internet. Nevertheless it does perform a useful role by describing, in detail, one of the UK's most successful early commercially available computers. It forms a valuable adjunct to the Science Museum's operational machine.

Godfrey N. Lance
ANU and Bermagui, NSW

MCCANN, D and THORNE, P. Eds. (2000): The Last of the First. CSIRAC: Australia's First Computer University of Melbourne Press, 176 pp., $30.00 (Paper Back)

There are two distinct parts to this book: The first called 'The CSIRAC Team' comprises eleven contributions from people who were involved with the design, construction, programming and maintenance of the computer. Each chapter describes the contribution made by an individual to the CSIRAC project. In seven cases the chapters are autobiographical; the remainder are written by friends or colleagues of the person in question. Clearly there is some overlap and repetition in these chapters but this adds to the interest rather than detracting from it.

The second part, of the book, comprises twenty more contributions from people who either used CSIRAC for their calculations or others who were involved with it. These papers were presented at a Conference, held in June, 1996 which was organised by one of the Editors (PT), to celebrate 40 years of computing in Melbourne. The very broad scope of the subjects in these chapters indicates what a valuable contribution CSIRAC made to scientific research in Australia in the 1950s and very early 1960s.

It is worth mentioning a few of the subjects to illustrate this point

i) Building research - structural problems and their application to the design of multistorey structures.

ii) Planning the Victorian Hydro-thermal Generating System.

iii) Atomic structure of metal surfaces.

iv) The production of solar position and radiation tables

In addition some computing for commerce and industry at financial institutions was also performed and this is covered in one of the chapters by Peter Murton.

CSIRAC was designed and built while Pearcy and Beard were working in the CSIRO Division of Radio physics and clearly this project was not one of the main stream activities of that Division. Hence the CSIRAC team felt some lack of support from management; however the fact that they persisted, and won in the end, is a great tribute to their hard work motivation and persistence. They were also disappointed that CSIRAC did not become the main stream computer for CSIRO. Rather commercially available computers were purchased in 1963 from Control Data Corporation and these were housed in a newly created Division of Computing Research within CSIRO. Experience, in many parts of the world, had shown that it was never satisfactory to use a research machine for routine calculations for too long. The designers, naturally,

were keen to 'improve' the machine and the programmers were frustrated whenever changes were made. All serious users wanted a stable platform for which they could write programs which would work for a long time without modification. Hence in this reviewer's opinion and with some hind sight the right decision was taken.

I read the whole of this book with considerable enjoyment and it is important for today's computer professionals to do the same. Note that CSIRAC is the only 'fully intact' computer left in the world today, which also has many fully operational programs available, so it has a unique place in the history of computing and everyone hopes this will long continue but obviously maintenance will become an ever increasing problem. Interestingly an emulator has been written for modern PCs and this can be used to simulate original CSIRAC programs.

I have one criticism of the book and this is that references and bibliographies are scattered throughout in a rather haphazard manner (they appear on at least six different pages) thus it is frustrating to find the reference which the chapter being read is talking about. I feel that the Editors could with little effort have improved the presentation and uniformity of the contributions by putting all references into 'The Bibliography' at the end of the book just before the comprehensive index. In this way the reader would always be able to read any reference without a deal of searching.

Godfrey N. Lance
ANU and Bermagui, NSW

NOTES

NOTES

Submission Guidelines

The Journal encourages submission of innovative and original articles in all areas of Information Technology including Computer Science, Software Engineering, Information Systems, Computer Systems and Information Engineering and Telecommunications.

The following forms of article are published:

- Full articles examining original research and/or application;
- Short articles describing original research and/or application or commenting on relevant legal, political or technical innovations;
- Survey or tutorial articles describing new directions in Information Technology or which are relevant to the practical application of fundamental research;
- Book Reviews;
- Announcements; and
- News Briefs.

Full details can be found at the Journal home page: www.jrpit.acs.org.au

Subscription Guidelines

The annual subscription for the Journal is

Individuals	
Australian Computer Society members	Free
Delivery address in Australia, incl. postage and GST when applicable	A$66
Overseas, incl. postage.	US$54
Institutions	
Australia, including postage and GST where and when applicable.	A$99
Overseas, including postage	US$81

All subscriptions to the Journal are payable in advance and should be sent (in Australian Currency) to the address below.

Advertising Information

The Journal accepts advertisements in the following categories:

- IT-related career opportunities
- IT-oriented training and education courses;
- IT-related conferences and workshops.

Rates are as follows:

Full Page	$1,500
Half Page	$750
Quarter Page	$375
By line	$100 for heading plus up to six lines of text (40 characters per line).
	$50 for each additional three lines, or part thereof.

Full details can be found at the Journal home page http://www.jrpit.acs.org.au

Bibliographic Information

Journal of Research and Practice in Information Technology – *J.Res.Prac.Inf.Tech.*
formerly the Australian Computer Journal (ISSN 004-8917)

ISSN	1443-458X
Circulation	15,500
Publisher	Australian Computer Society Inc. PO Box Q534, QVB Post Office Sydney 1230, New South Wales AUSTRALIA
Production Management	Associated Business Publications Pty. Ltd. P.O. Box 3267, Umina Beach, NSW 2257, Australia
Cover Design	Modern Planet Design (08) 8340 1361 • modern@webmedia.com.au

Contents

194 On Smoothing Algorithms for Transmission of Stored MPEG Video
F. Li and Y. Liu

210 MAT: a Mobile Agent System for Supporting Autonomous Mobile Agents
W. Li and M. Zhang

227 A Meta-data Update Policy in a Flash Memory File System
S. Cho, J. Lee and J. Ma

239 The Effects of Normalisation on the Ability of Business End-Users to Detect Data Anomalies: An Experimental Evaluation
A. F. Borthick, P. L. Bowen, M. R. Liu and F. H. Rohde

Short Communication

263 The Code Red Epidemic: a Case Study
J. Lamp

Special Features

193 Editorial John Roddick

267 Book Reviews

A publication of the Australian Computer Socie
ISSN 1443- 458X